DOCUMENT RESUME

ED 281 482 IR 012 593

AUTHOR Ballay, Joseph M.; And Others
TITLE CMU/IBM Usability Study: Final Report. CDC Technical Report No. 11.
INSTITUTION Carnegie-Mellon Univ., Pittsburgh, PA. Communications Design Center.
REPORT NO CDC-TR-11
PUB DATE Mar 84
NOTE 112p.
PUB TYPE Reports - Research/Technical (143)

EDRS PRICE MF01/PC05 Plus Postage.
DESCRIPTORS Communications; *Computer Assisted Instruction; *Computer Software; Design; Engineering; Higher Education; Industry; Systems Development
IDENTIFIERS *Computer Assisted Design; *Computer Users; Design (Engineering); Information Management; Visual Representation

ABSTRACT

This report focuses on the activities and findings of the fourth phase of research carried out at Carnegie-Mellon University (CMU) on the development of user documentation for computer-aided design (CAD) systems. The first of four major sections provides an overview of recent research, issues involved in the research, and implications of the findings that will enhance the design process as it is practiced by engineers and industrial designers. The second section reports on user process research, which involved the observation of approximately 50 designers, design students, engineers, and draftsmen. Topics addressed include the character of the design process, selective refinement of visual images created and used in the process of problem solving, and techniques used to manage the large amount of external information used by the designer in reaching a final solution. Several concepts that should influence future development of computer-aided design software manuals and computer manuals in general are identified in the third section, which discusses the development of the second module of the CMU CADAM Manual for IBM (International Business Machines). Computer-assisted instruction (CAI) for CADM design software is compared to printed documentation in the fourth section. The appendix includes background on the development of documentation for CADAM, CAI program documentation, and CAI testing materials. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

CDC Technical Report Series

CDC Technical Report No. 11

CMU/IBM Usability Study: Final Report

Joseph M. Ballay, Karen Graham, John R. Hayes, and David Fallside

March 1984

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Communications Design Center

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Thomas M. Duffy

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

The Communications Design Center was established in 1979 to solve communication problems in business, industry, government, and the professions.

The goals of the Center are:

- to encourage interdisciplinary approaches to the solution of visual and verbal communication problems;
- to promote cooperation between communication researchers at the University and those outside the University who have practical interests in communication problems;
- to provide resources and an organizational setting for research in visual and verbal communication;
- to train professional writers; researchers, and teachers;
- to promote the development of educational materials for communication programs on and off the CMU campus.

This report is one of a series from our on-going research on communication and document design. If you have questions or comments on this report, or if you would like to find out more about the activities of the CDC, contact:

Director
Communications Design Center
Baker Hall 160
Carnegie-Mellon University
Pittsburgh, PA 15213
(412) 578-2906

Final Report
Phase IV

March 1984

CMU/IBM Usability Study

Joseph M. Ballay
Karen Graham
John R. Hayes
David Fallside

Carnegie-Mellon University 1984

TABLE OF CONTENTS

Overview and Implications 4

User Process Research 10

The character of the Design Process 10

Selective Refinement 22

Information Management 32

User Documentation - Module Two 36

Module Two - Research and Implications 36

User Aids - Computer-Aided Instruction 45

CAI/ CADAM 45

Appendix:

Appendix A: 66
Module One and Two Background

Appendix B: 81
CAI Program Documentation

Appendix C: 97
CAI Testing Materials

Overview and Implications

This final report covers our most recent research and summarizes Phase IV. It also lists implications for the design of computer systems that will enhance the design process as it is practiced by engineers and industrial designers. Every part of the Phase IV effort – the User Process Research, Module Two of the tutorial, and CAI/CADAM – contributes to these implications. The User Process Research section offers implications by considering three information-processing analyses of the design process. The Module Two and CAI/CADAM sections offer implications from the experience of writing, designing and programming tutorial documentation for an existing system, CADAM.

The User Process Research section is based on the observation of approximately fifty designers, design students, engineers and draftsmen. Of these, eighteen were the subjects of detailed protocol analysis and one subject was studied in-depth involving approximately twenty hours of videotaped protocols. In this section the user is called "subject" in reference to evidence based on a specific protocol, and is called "designer" in reference to more general observations.

User Process - Design as a cooperative game.

At the broadest level, the designer/client level, the design process may be viewed as a cooperative game between designer and client, a game in the sense that the participants take turns. It is cooperative because both designer and client help each other in working toward the same goal – producing an excellent product. The client tries to help the designer by providing accurate information about the product's function, the allowable cost, decisions about alternatives, etc. The designer, in turn, tries to help the client by providing clear information about the available design options. The designer and client continue to alternate turns until the designer's part in the development of the product is complete – say, when the design can be turned over to a manufacturer or a model maker.

The reason that the design process proceeds by turns may be that designers and clients have very different competencies. Clients typically know their own resources and the nature of the market for the proposed product better than designers do. Designers, on the other hand, typically know the design process far better than clients. Therefore it is efficient for each to work independently, meeting to give or receive information and feedback.

User Process - Design as a sequence of representations.

There are professions in which the core professional activity involves solving problems based on information which is embedded in visual representations. Industrial design and many kinds of engineering such as circuit layout and package engineering fall into this category. Our studies indicate that people who work on these kinds of problems rely heavily on visual imagery; not only the formalized images of draftings and diagrams but informal images – sketches – which they create and use in the process of problem solving. The creation of representations is a visual and tactile act that proceeds by accumulating inventions and design decisions at increasing levels of precision; a process we call selective refinement. Unfortunately, these two aspects of selective refinement – the physical interaction between a designer and his sketch, and the gradual increase in precision – are often not provided in automated design systems.

User process - Design and Information Management.

The designer considers a large amount of information while generating a design. This information is of many types, e.g. cost, color, and size, and it is also present in many different forms, e.g. scale drawings, text, and photographs. The designer is faced with the problem of how to manage all this information. For example, if the designer does not store information efficiently then it is likely to be lost. Then the designer will have to regenerate the information or make a decision with incomplete information. If the designer's organization of information is poor then he will have to spend extra time searching it. Regenerating and searching information in a design task are particularly costly because of the quantity of information involved.

We found that our subjects used a few simple techniques for managing information. Two of the techniques involved using different locations to separate information. In one case the locations were permanent – dividing the workspace into different areas. In the other case, information was assigned to temporary locations. A third technique involved mixing information between representations, e.g. adding cost information to a perspective drawing.

The last technique was a labelling scheme which allowed the subject's to quickly reference collections of objects. Using these techniques, the subjects were able to store and retrieve information with a minimal amount of effort.

User Documentation - Module Two

Module Two was supported jointly by the Kingston Design Center and the MDA Design Practices group in Poughkeepsie. The goal of Module Two was to develop a comprehensive basic training manual using the same strategies implemented with Module One. Module Two is the second part of a two module set. Two additional support documents, the Reference Summaries and the Pocket Reference Guide, were written to complete the set.

As a result of our work on Module Two we have identified several concepts that should influence future development of Computer-Aided Design software manuals and computer manuals in general. These concepts derive from the Protocol Aided Revision process which helps to clarify the interaction between the user, his documentation, and the system. Concepts such as audience definition, document layering by audience expertise, information access routes, information density and symbolic cues can be seen in relation to the design of whole systems as well as the design of documentation alone.

User Aids - Computer Aided Instruction

Because comparison of CAI to printed presentation was the major goal, CAI/CADAM intentionally parallels the large scale organization of the CMU CADAM tutorial Module One. That is, a short introduction followed by a series of sequential exercises including reference summaries at the end of each exercise, then a glossary and a help section. The CAI medium, however, presents redesign opportunities for the presentation of information on screens and the access to that information via menus, scrolling, and contextual locators. The prototype software demonstrates the potential of CAI documentation as well as difficulties to be overcome. A major by-product of CAI/CADAM is a demonstration of the benefit that graphic conventions can have on the usability of CAI documentation.

Implications for Automated Design Systems

The multiple views of usability mentioned above have yielded a number of implications for automated design systems. They are summarized here in relation to five major issues that we encountered in our research. It should be understood that this summary arrangement is for the purpose of simplification because most of the issues overlap. Using these implications, we will be describing an advanced design automation system in more detail in the first quarterly report from Phase V of our research. This report will be published in June 1984.

Broad level user processes These processes include consideration of designing as a cooperative game, the range of specific processes included in designing, and recognition that various subsets of these processes pertain to different kinds of design problems and users with different abilities.

- An automated design system is also a communication link among the engineering and design team who are solving a problem, and with the management or clients who oversee the problem.
- A complete system will require the interaction of many specialized application programs. This implies compatibility of data between programs so they can be assembled into packages which satisfy users in different problem domains, or users with different levels of expertise. One can imagine a manager's set, an industrial designer's set, a packaging engineer's set, etc.

Information Management This issue is concerned with the designer's methods for organizing the large amount of information that is typically used in a design task.

- Much of information management has a spatial component; organization is achieved by putting information in different locations.
- The designer uses a minimum of three areas that differ in the visual accessibility of the information they contain, 1) decision-making work space, most accessible, 2) display/reference space, less accessible, 3) mass storage, not visually accessible.
- Partitioning of information (especially of visual information) and immediate labeling of parts is essential to create identifiable units from which design solutions are assembled. It should be possible to automate the labeling process.
- Designers work with categories of information that are largely predictable. This should permit automated filing of labeled information.

User Interface The interface includes manuals and other user aids as well as components of the Automated Design System - screen images, control language, input controls, etc.

- Manuals and other user aids should be specified early and lead in defining other aspects of the interface specification.
- The user must be aided in building a conceptual map of the whole system. The process is incremental and personalized between individual users.
- A major contribution to the conceptual map is alternative and clear pathways to a particular piece of information (e.g. alphabetically, spatially, or by position on the solution path).
- For visual problem solving, the user's primary contact with the visual information should be through the image itself; not through alphanumeric abstractions, not even through menus.
- On-line *reference* information about system operation for occasional users is particularly advantageous, on the condition that it is well written and appropriate for the software. The visual presentation of the information should also be handled using design principles developed for on-line presentation.
- On-line terminology and symbols must be concise and consistent across such application programs as text editing and solid modelling. Interfaces designed to the same conventions will enable the details of the user process to be compatible across the modules.

Multiple Representations Designers and engineers use many modes of representation in problem solving and switch between them frequently. A minimum set of representations includes: procedural representations, solid models, matrices, orthographics, perspectives, alphanumerics, and dimensions.

- Automated Design systems will have to accommodate this minimum set of representations if the user is not to be forced off-line for major parts of the design process.
- Furthermore, the designer should be allowed to mix these representations. For example, annotating solid models with alphanumerics.
- Special importance will have to be placed on providing represention of solid models. This is not a matter of the designer's preference but a real cognitive need to "get the feel" of objects.
- The designer also transfers information between representations, for example between perspectives and plan views, and so this type of operation should be provided.
- The translation of information into another representation does not have to be instantaneous (on the order of a few seconds is probably acceptable).

Changing task environment The design process proceeds by a series of information transactions between the user and his environment (including computers in his environment). The information transactions are recorded as a series of representations which are a part of the task environment.

- The criteria in a design problem are interrelated and change. Automatic monitoring of criteria will be useful in the early phases of the design process, for example during criteria formulation, and also later on for evaluating proposed solutions.
- There are variations in problem solving processes that respond to the requirements of different problem domains and different individual styles of working. A personalized management system can anticipate particular representations that will be needed.
- A personalized management system would record a trace of the representations used for a particular problem. The trace could be analyzed to detect looping indicated by partial solutions which are frequently revisited. The trace could also be played back to reacquaint a designer with his previous decisions.
- The designer should be able to specify the precision, thus allowing himself to work at an economical level of refinement, especially in the early representations.
- Image processing should allow for: 1) extracting information out of one representation as the basis on which to build the next representation, and 2) sensing dissonances in precision, inclusion, or coherence between views in the same representation.

The remainder of this report will elaborate on the research conducted in Phase IV of the CMU/IBM Usability Study.

User Process Research

The Character of the Design Process

Design as a cooperative game

At the broadest level, the designer/client level, the design process may be viewed as a cooperative game between designer and client. It is cooperative because designer and client help each other in working toward the same goal -- producing an excellent product. The client tries to help the designer by providing accurate information about the product's function,the allowable cost, decisions about alternatives, etc. The designer, in turn, tries to help the client by providing clear information about the available design options.

At this level, the design process may be viewed as a game in the sense that the participants take turns. The first turn is taken by the client who proposes the original project. The second turn is taken by the designer who develops a set of alternative concepts and proposes them to the client. In the third turn, the client selects one or a few of the concepts and suggests modifications or directions for further development. The designer and client continue to alternate turns until the designer's part in the development of the product is complete -- say, when the design can be turned over to a manufacturer or a model maker.

The reason that the design process proceeds by turns (and it very commonly does) may be that designers and clients have very different competencies. Clients typically know their own resources and the nature of the market for the proposed product better than designers do. Designers, on the other hand, typically know the design process far better than do clients. It is efficient, then, for each to work independently, meeting only when one needs to give or receive information from the other.

Planning by turns

The turn-taking character of the design process has important implications for the way the designer plans. The fate of an idea proposed by the designer in one turn is often determined by the decision of the client in the next turn. For this reason, designers often limit the amount of effort they invest in an idea until the client has had a chance to review it.

For example, in his first turn, the subject in the in-depth protocol, proposed four concepts but did not develop them in detail because he anticipated that most would be rejected by the client at the next turn. Thus, to an important extent, the designer's planning is limited in scope to a single turn. We describe this phenomenon as "planning by turns."

Design as an ill-defined construction task

While it is useful to characterize the design process as an interaction between designers and clients, it is also useful to examine design from the viewpoint of the designer as an individual problem solver. Two aspects of design tasks have important impact on the way design problems are solved, they are:

1. design tasks are construction tasks, and
2. design tasks are ill-defined.

By a construction task, we mean a task which has as its main goal the creation by cumulative action of a persisting external product -- a product which must satisfy a demanding set of criteria before the task will be judged as having been successfully completed. Written composition is an example of a construction task because its main goal is the production of an external object, the text, which must satisfy criteria of relevance, clarity, organization, effectiveness, etc. Industrial design is an example of a construction task because its main goal is the production of an external representation of the solution to a design problem which must satisfy criteria of utility, clarity, esthetics, etc. Other examples of construction tasks are computer programming, carpentry, sculpting, and constructing a HEATHKIT amplifier.

Construction tasks vary widely in the demands they place on the problem solver for decisions about the form of the final product. Tasks such as routine carpentry and HEATHKIT amplifier construction place minimal demands on the problem solver for decisions about the nature of the end result. In contrast, tasks such as sculpting and computer programming impose major demands on the problem solver to contribute to the shaping of the final product. Tasks such as these last two are described as ill-defined (Reitman, 1965) because they do not provide the problem solver with sufficient information to specify the final product. Problem solvers, therefore, must provide information through their own decisions which further specify the task and define the form of the final product. Writing and industrial design, clearly, are also examples of ill-defined construction tasks.

There are a number of consequences which follow if a task is an ill-defined construction task:

1. The partially completed product becomes part of the task environment. That is, what has been constructed up to now influences the subsequent course of problem solution and the shape of the final product. In writing, the writer frequently rereads the text produced so far to provide context for the next segment of text to be produced. Kaufer, Hayes, and Flower (1983) have shown that in composing sentences, writers frequently reread the beginning of an incomplete sentence in order to get a 'running start' to complete it. In design, the designer's current sketches are used as a source of ideas and inferences which shape later design decisions. For example, in one design episode, we observed a designer examining a crude model of a proposed solution and discovering that it would be easier for right handed people to use it than left handed people. The designer was alerted to the issue of handedness which then became an important criteria by which to judge later solutions.

2. Because the partially completed product is part of the task environment and because it is continually changing, the problem solving environment is continually changing in ill defined construction tasks. These changes in environment stimulate new ideas and inferences. As a result, invention is stimulated continuously throughout the course of the solution of ill defined construction tasks. Indeed, we have observed both in writing and design tasks that invention occurs continually in these tasks right up to the moment when the final draft or final drawing is complete.

3. Because construction tasks produce a final product which must satisfy external criteria, such tasks necessarily involve a commitment of resources. In writing, one must eventually invest the energy to transform crude ideas into formal sentences on paper. In design, one must produce a clear presentable drawing. In carpentry, one must eventually cut and shape the wood. The more expensive the commitment of resources is in producing the final product, the more desirable it will be to plan. Because construction tasks all involve commitment of expensive resources, all benefit from planning. We have observed that planning is an important part of writing (Flower and Hayes 1981) and of the design process (Ballay, Hayes, Fallside, and Graham 1984) especially in expert writers and designers.

Planning reduces the effort involved in carrying out a construction task because it reduces the cost of errors and false starts made in searching for the best solution. Architects don't want to explore in steel and concrete. They want to have a plan before they commit themselves to expensive materials. In ill-defined construction tasks, then, we expect that early in the process of solution, problem solvers will work toward solution using relatively inexpensive representations -- rough sketches, notes, etc., and later, work with progressively more expensive representations -- e.g. more precise drawings, formal text, etc., before committing themselves to a solution in the most expensive representation

-- the external product. This progression from crude, inexpensive representations to more precise and expensive ones is evident in the in-depth protocol. It will be discussed in considerable detail later.

Design is a visual task

As we have seen, design resembles other ill-defined construction tasks such as writing in a number of important ways. However, there are also some important ways in which design differs from other ill-defined construction tasks. Perhaps the most important has to do with the "visual" nature of design. If a designer has several ideas in mind, he or she typically can't work out the relations among them until they are put down on paper. An external drawing is a very important aid to the designer in making spatial inferences. Experienced design teachers find that students who claim to have designed wonderful things "in their heads" typically discover disastrous flaws in their designs when they try to put them on paper. Experienced designers emphasize the importance of the external representation in the process of graphic thinking.

Types of Representation in Design

The subject in our in-depth study makes use of seven distinct types of representation in the course of solving the design problem. These are:

Procedural Representations
Solid Models
Matrices (Symbolic/Graphic I)
Orthographic projections
Alphanumeric Representations
Perspective Drawings
Dimensions (Symbolic/Graphic II)

Each of these representational types appears to serve some unique functions in aiding the design process.

Procedural Representations

Procedural representations encode action scenario's. For example, they are used to describe how the user will operate the Easy Banker. Procedural representations take the form (goal ((cue - action)* result)*. The asterisks indicate that the element in parentheses may be repeated. The examples below, taken from the protocols, illustrate this form.

Figure 1

Procedural Representation

[Goal] –	"I bought something with my credit card...now it comes time that I have to pay my monthly bill on that purchase..."
[Action, cue implicit] –	"Okay, so I put my credit card in with my left hand..."
[Response] –	"It identifies me."
[Cue] –	"It says, "Identify yourself".
[Action] –	"This is Tim Cunningham."
[Response] –	"and the voice is recognized."

Solid Models

Solid models serve at least three functions:

1. to help the designer perceive volumetric relations.

2. to facilitate the arrangement of volumes with respect to one another in space, and

3. to orient volumes in space

One of the most striking features of the protocols was the subject's insistence that he needed solid models to perceive volumetric relations. Describing the solid models, the subject says,

> "when you look at them on the drawing, you tend to focus just on the key area and not notice just how much dimensional difference there is, overall, to the component."

In fact, the subject appears to discover some volumetric relations from the solid models which were not apparent to him from the drawings provided. At one point, he says

> "it's important for me to get a sense of what these really are and I didn't notice these from just looking at the flat drawing, as to how dimensionally different in volume these things are – that the keys on one stick up much higher than the keys on the other and the pins stick out."

A second function of the solid models is to facilitate the arrangement of volumes in space. For example, the designer says,

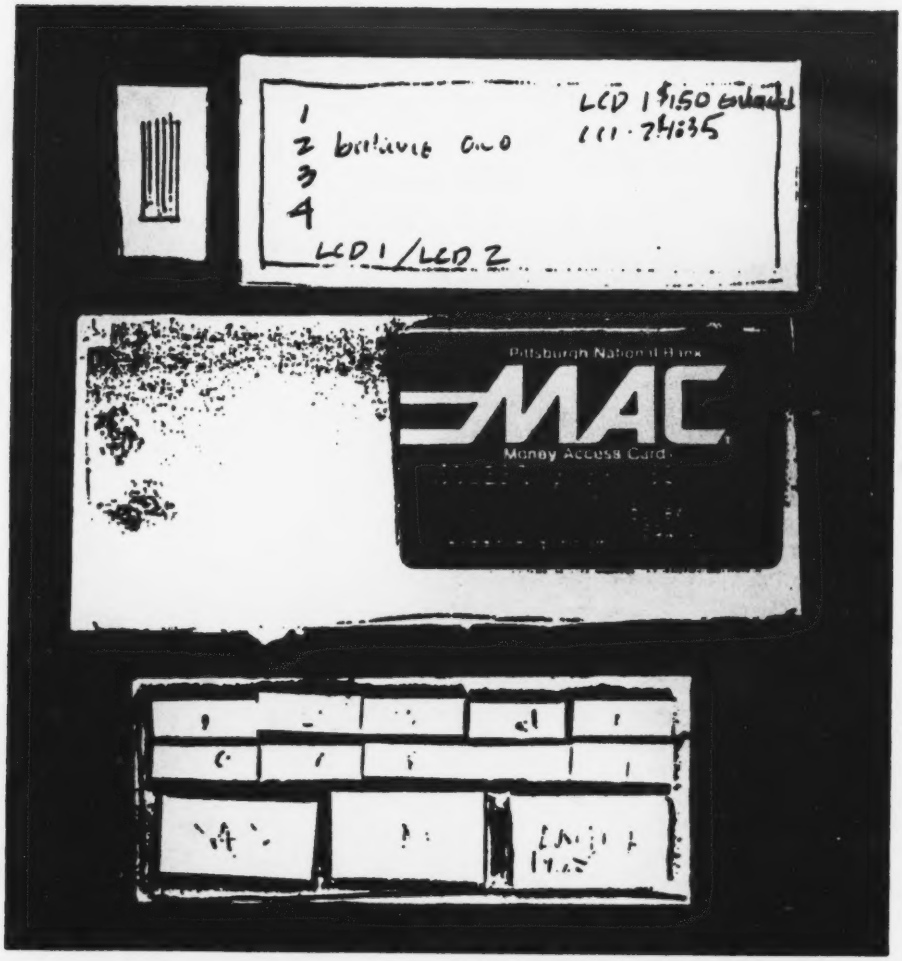
> "So I can do these quite quickly and roughly pile them around and move components around and come up with several configurations..."

In fact, the four configurations which the subject produced were arrived at by rearranging the solid models on the table top.

The third function of the solid models was to assist the subject in specifying the orientation of components in space. For example, while the subject was drawing a perspective sketch, he decided that the display might be tilted. To decide on the angle of tilt, he held the solid model of the display next to the drawing and tilted it at various angles to observe the effect.

Figure 2

Solid Model example from the protocol



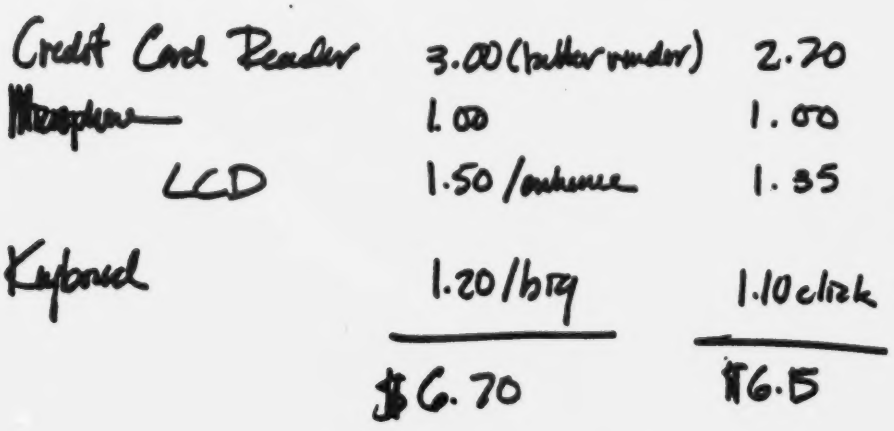
Matrices

When he finished cutting out the solid models of the part, the subject constructed a matrix to help him to choose which of the alternative parts he wanted to work with. The rows of the matrix represented part types, e.g., card readers, microphones, etc., and the columns represented alternative models of each type. Information concerning each model is contained in the cells. Thus, keyboard #1 costs $1.20 and is "big" while keyboard #2 costs $1.10 and clicks when the keys are pressed.

The matrix, frequently used as a decision making aid, is a type of representation which we have called Symbolic/Graphic I. This representation consists of a symbolic component – the alphanumerics in the matrix – and a graphic component – the structured array in which the symbols are located. In Symbolic/Graphic I, the graphic component facilitates the retrieval and comparison of information but is not essential to the meaning, i.e., the matrix could be arranged in many different ways without changing the meaning of the information it contains. Later we will discuss a representation we call Symbolic/Graphic II in which the graphic component is an essential part of the meaning.

Figure 3

Matrix example from the protocol

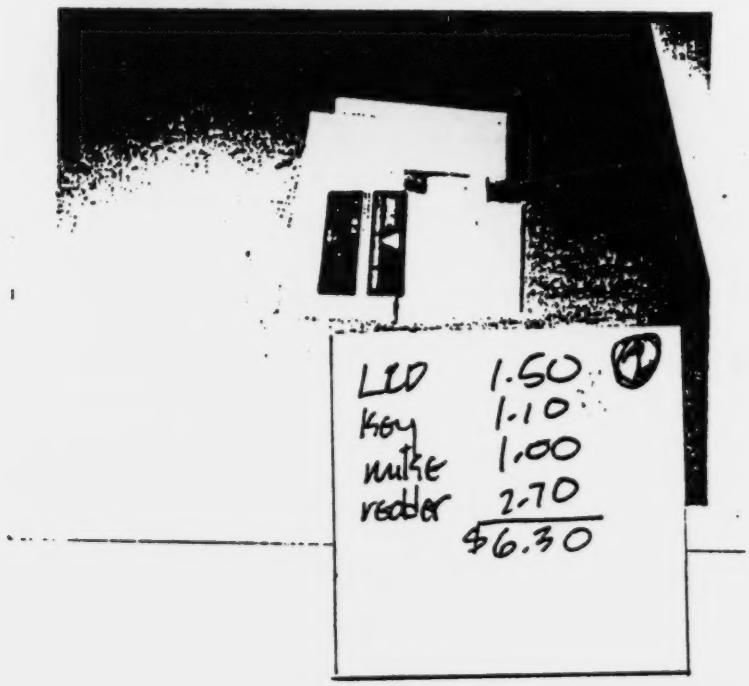


Orthographic projections

As the subject completed each arrangement of solid models, he took a polaroid photograph of it. The photograph provided a crude plan view which was used as a record of the locations of the parts with respect to each other. Since the parts were photographed lying flat on a table, the picture contains no information about tilt. For example, in some arrangements, the subject intended the display to be tilted at 45 degrees, but this information is not contained in the photographs.

Figure 4

Orthographic projection in the form of a polaroid



Alphanumerics

After taking a photograph of an arrangement, the subject attached a label to it (See Fig 4) which specified the arrangement's number, cost, and parts. The form of representation was alphanumeric – frequently used for description and labeling.

Perspective Drawings

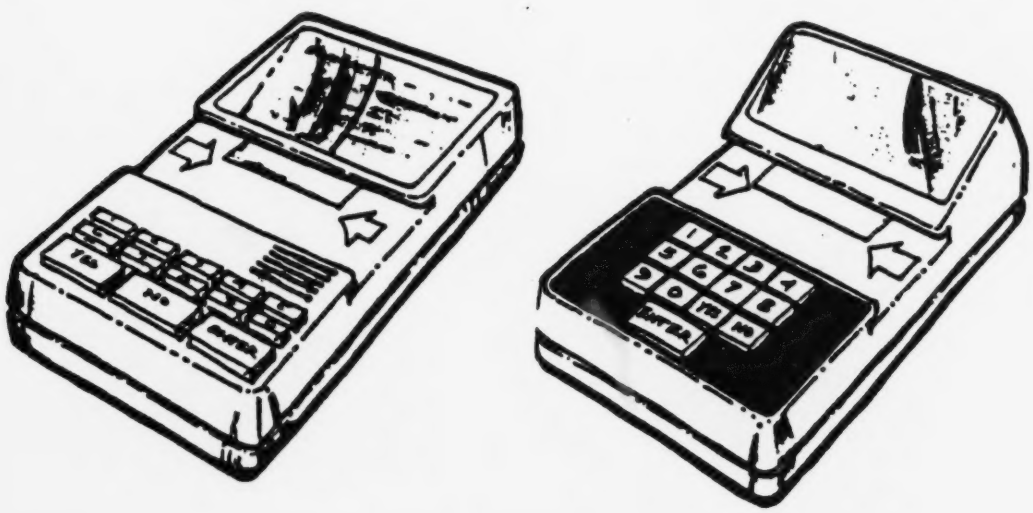
After the subject had generated and photographed four arrangements, he drew a crude perspective sketch of each. These perspective sketches serve to record the subject's decisions about the orientation of the parts and about the appearance of the envelope that contains them. In addition, the act of sketching appears to be an important occasion to make decisions on these and other issues. For example, while sketching Concept #3, the subject decides to recess the LCD display, and while sketching Concept #1 he considers a plexiglass cover for the card holder.

The subject intends these rough sketches just for his own use. He says, "they are just in this form which you understand and they make some sense to you but you could never sell these to anybody." The subject requires much more complete sketches for presentation to the client.

The presentation drawing consists of carefully constructed orthographic and perspective views. It's purpose is to present the subject's concept to the client in as attractive and intelligible a form as possible. However, as with the rough sketches, constructing the presentation drawing was an occasion to make improvements in the design. For example, while drawing the plan view, the subject decided to add arrows to show the user where to insert the credit card.

Figure 5

Perspective Drawing example

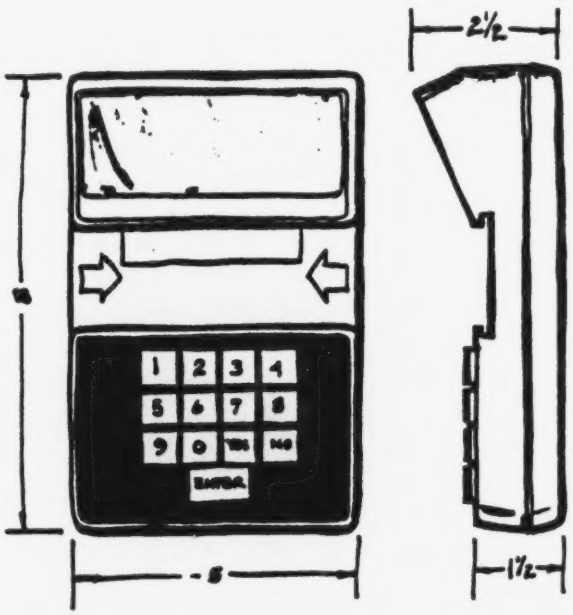


Dimensions

After completing each sketch, the subject dimensioned them. A dimension is a representation which consists of a combination of symbolic and graphic components. In this representation, which we have called Symbolic/Graphic 11, meaning is conveyed both by the symbolic and by the graphic component. If we changed either the symbols or the positions or lengths of the arrows, we would change the meaning of the representation.

Figure 6

Dimensioned Drawing

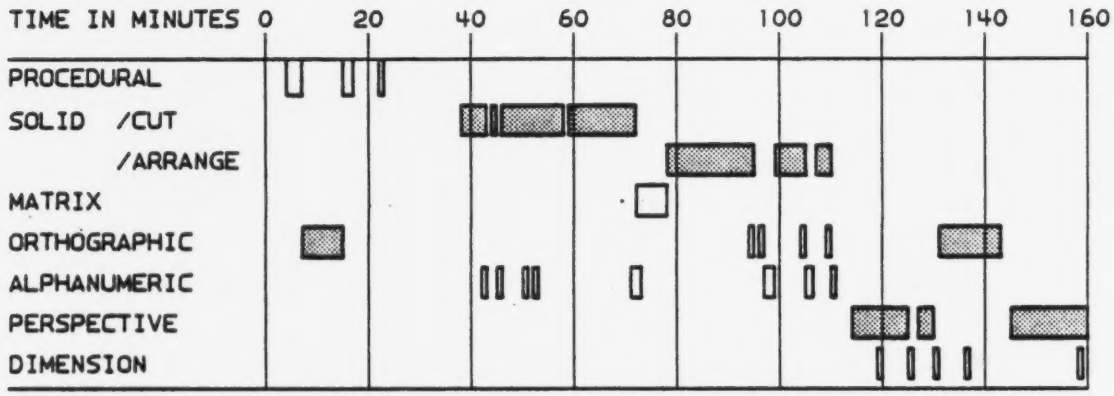


Representational Episodes

It is evident in the design protocols that the subject is using the various representational modes episodically. That is, he tends to do all the work he has to do in one representational mode at one time. When that work is done, he switches to another representational mode and does a block of work in that representation. Figure 7 shows these episodes graphically.

Figure 7

Representational Episodes



At this point, it is not clear to us why the subject uses representations in this episodic way. Understanding this phenomenon might help us to understand how designers organize their work. The episoding phenomenon has some implications for the design of computer-aided design systems. Since designers switch between representational types relatively infrequently, the merit of a design system may not depend critically on the cost of switching between representations.

Conclusions

The special properties of the design process which we have described here have important implications for the designers of automated design systems. As we have noted, design is a visual task in the sense that much of the thought and discovery in the design process is mediated by external visual representations. This makes it especially important that automated design systems allow the designer to construct representations easily and quickly. In addition, it should assist them in keeping track of what may be a rather large number of such representations (we have observed that designers working with paper and foam-board frequently lose things on their desks.)

Some of the representations which designers use are relatively easy to implement on computer systems. Orthographic projections and perspective drawings are examples which are most obviously useful to the designer. Less obvious but also very useful to the designer would be a package to assist in the construction of matrices. (Designers seem less comfortable than engineers in constructing matrices and may be especially grateful for the help.)

One representation appears to create special problems for automated design systems -- solid models. Subjects express a strong desire to use solid models in the design process. This desire is not just a matter of preference on their part but seems to reflect a real cognitive need. Designers discover things from solid models that they do not discover from orthographic projections which "contain the same information." The solid modeling programs currently available in design systems do not appear to be thoroughly satisfactory to designers. Perhaps this is because, as the subjects say, they can't "get the feel" of the objects. Perhaps it is because the parts in such representations can't be manipulated independently, or the software is too difficult to learn.

Whether automated design systems can provide solid models or some alternative representation which can serve the same functions is a question that the designers of such systems must consider most seriously. Any design system which fails to provide a representation which serves the function of solid models is not likely to be popular with designers.

Another very important consideration in designing automated design systems is the issue of imprecision. As we discussed above, economy in planning often requires the designer to represent information imprecisely -- to sketch. Designers want to sketch and there are good cognitive reasons for them to do so. Systems which don't allow sketching won't please designers and won't promote good design.

Selective Refinement

Introduction

There are professions in which the core professional activity involves solving problems based on information which is represented in visual images. Industrial design and many kinds of engineering such as circuit layout and packaging engineering fall into this category. Our studies indicate that people who work on these kinds of problems rely heavily on visual imagery; not only the formalized images of draftings and diagrams but informal images – sketches – which they create and use in the process of problem solving. The creation of these sketch images is an intensely visual and tactile act. These people "get into their work" whether it be at a blackboard, on the back of an envelope or at a graphics computer terminal.

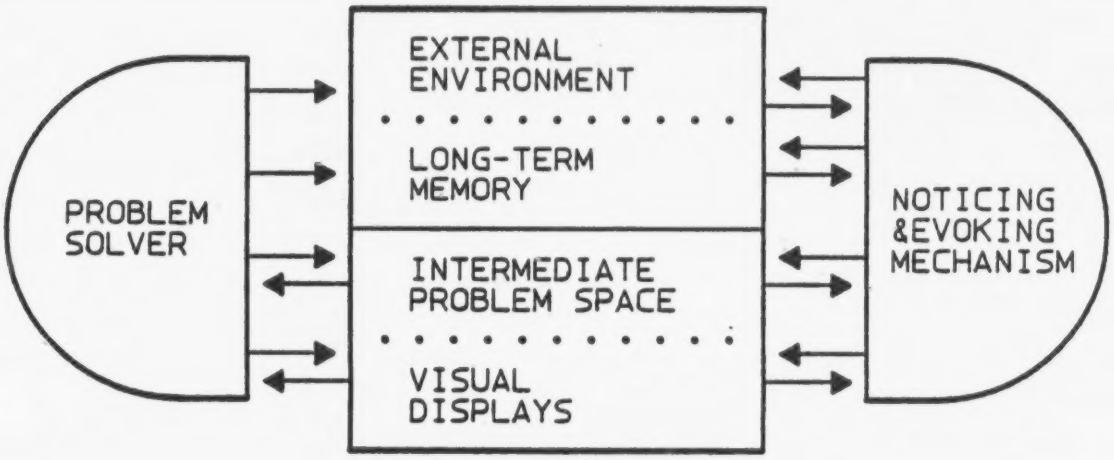
Our principal concern has been the computer environment and its affect on sketching – the most creative part of the visual problem solving process. An immediate observation is that graphics hardware and software seems to inhibit creative sketching rather than enhance it, apparently by contradicting the intimate visual/tactile connection between the sketcher and his sketch. We studied information contained in our design process protocols for indications of the fundamental relationships between the sketcher, the information contained in the sketch, the process by which the sketch is created, and the environment (computer based or paper based) in which one sketches. We also classified the various sketching activities.

Theoretical Models

The prevailing model of human problem solving is the information processing model proposed by Simon (The Structure of Ill-structured Problems, 1973). It is a model that is very successful at explaining how problems are solved by identifying the kind of information that has to be accounted for, how and where that information is kept, and describing some basic processes that change the information and move it around.

Figure 8

Model for Ill-structured Problems



One of the attractive qualities of the model is that it can describe the way people solve ill-structured problems. Within this framework, Hayes describes some common tasks that people do, like expository writing and industrial design as ill-structured construction tasks. "Ill-structured" means that the solution is incompletely described and that no methods guarantee an acceptable solution. "Construction" means that a persisting external product is created in the task. The interesting thing about the ill-structured construction task concept is that it implies some theoretical problem solving strategies that are important to understanding how sketching fits into the overall design process:

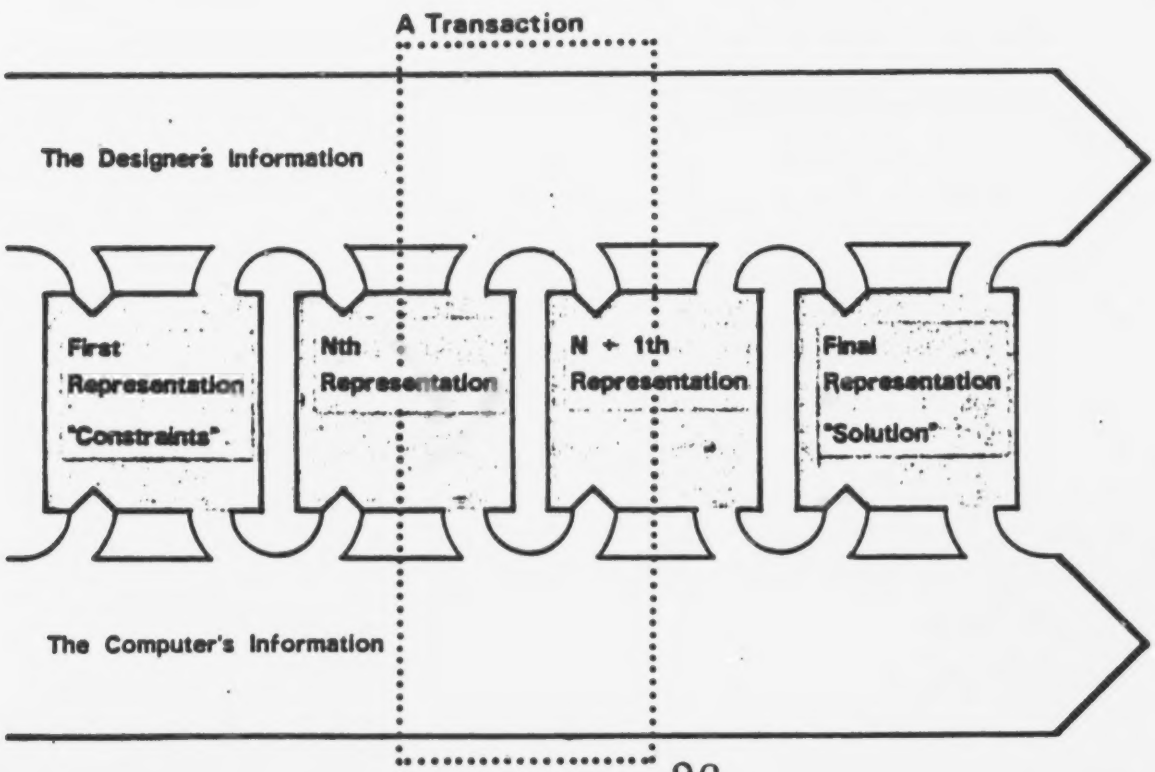
- short episodes of staged planning
- alternating with periods of routine production
- generally increased commitment in subsequent cycles
- and invention continues throughout the design process

For example, the continual invention is dependent on something like the noticing and evoking mechanism shown in Figure 8. Protocol evidence presents many instances in which an invention occurs out of sequence from the issues that are being considered immediately before or after the invention. The subject is studying the sketches that have been made and notices an opportunity to solve a problem that had been lurking at low priority until then. In one instance a solution for providing additional cooling derived from the opportunity to enlarge a grill area that had already been sketched for locating a microphone unit. He may also suddenly invoke a constraint or design standard that had not previously been a part of the immediate problem space. This occurred when a sketch model, originally intended to generate alternative component assemblies, indicated that one layout was biased against left handed users.

At the procedural level, the design process can be explained by cyclical model shown in Figure 9. It is useful in describing where information comes from and goes to at any stage of the design process. It accounts for the information that is held in the designer's head and the information that is held in the external environment, held in a computer in the case of an automated design system. Most important, it puts representations at the center as the medium of exchange between the designer and the computer.

Figure 9

Transaction Diagram



We have noted that subjects don't know many details about what they are going to sketch until part of the sketch is made; "knowing" is in the observing of the external sketch. According to the information transaction model, (Figure 9), a designer begins sketching (enters the transaction) with only partial information about the sketch that will be produced. Additional information is retrieved from the environment – including computers in the environment. A record of the information processes is encoded in the representations and subsequently stored in the designer's memory, and his environment as well. The representations are external and can take a variety of modes. In one subject's protocol we counted eight modes of representation: Procedural representations, Solid models, Matrices, Photographs, Symbolic representations, Perspective views, Orthographic views, and Dimensions.

All of these can be produced in what we would call a sketch, that is a less refined form. If one imagines putting these information transaction processes end to end, the external artifacts of the design process would appear as a chain of representations, the majority of which are sketches. The number of sketches produced in the protocols indicates the importance of sketching to the creative design process.

Evidence from actual sketch output: the dimensions of sketching

The traditional, paper based design process produces many artifacts, most of them sketch representations of partial solutions. We examined the sketch output from our design scenario along with the videotaped protocols of the sequences in which these sketches were produced. As a result of this examination we identified three variables or dimensions of sketching which help describe the process by which information is put into sketches. This process is broader than the traditional view of sketching as sloppy drawing and has been given the name "selective refinement".

Transaction/Inclusion - Information From Previous Sketches

Most new sketches don't start from scratch. In keeping with the idea of the information transaction, it builds on information that is extracted from previous sketches. In paper based systems it is common to use tracing paper to extract information in an obvious, external way. We also have evidence that it happens at the cognitive level. In one protocol segment, a subject sketched four alternative component arrangements: the first took 13 minutes, the others about five minutes each. Clearly some information, in the form of partial solutions, was being extracted from the first alternative and being used in the succeeding arrangements. There was a decision about inclusion.

Inclusion is one of the three dimensions – inclusion, coherence, and precision – that describe the ways in which sketches differ from "finished work".

> *Inclusion* is the amount of information about a form that is represented in a sketch. It might be understood as the level of detail or as the "grain size" of the information in the sketch.
>
> *Coherence* is the degree to which different pieces of information agree with or support one another. It reflects whether the partial solutions to sub-problems have been reconciled to one another.
>
> *Precision* is the most commonly understood of the dimensions. It is the refinement with which an intended configuration is represented.

While planning, a designer makes several choices. He chooses a representation and levels of precision; inclusion and coherence. The choices are made, based on training and experience, to be appropriate to state of the problem as he understands it at that time.

The choice of a representation is obviously an important decision. Perspectives show information that numerical dimensions don't and vice versa. That is why a completed sketch is often a combination of modes. The decisions about inclusion, coherence and precision are involved in the design process in other ways which will be discussed. The important point is that for experienced designers, these variables were under the subject's control. The looseness of a sketch, for instance, is a conscious choice on the part of the subject, not simply the result of being sloppy or in a hurry.

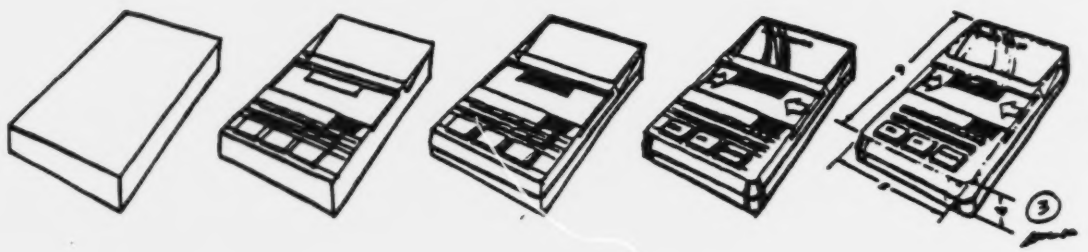
Inclusion - Adding Information To A Sketch

Clearly a sketch can be more inclusive as it is worked on. Interestingly, the information in a sketch seems to get added in a consistent sequence that is partly the result of a designer's training and experience, and partly a response to the demands of the particular design problem. This implies that computer supported design might include a personalized expert system that responds by adjusting to specific problem types or to specific users. One subject, in every orthographic and perspective sketch he produced, added information in the following sequence:

- Proportion of envelope (product housing)
- Component locations
- Mechanical and Aesthetic refinements
- Material & surface attributes (surface reaction to light)
- Metrics (notations and dimensions)

Figure 10

Sequence of Inclusion



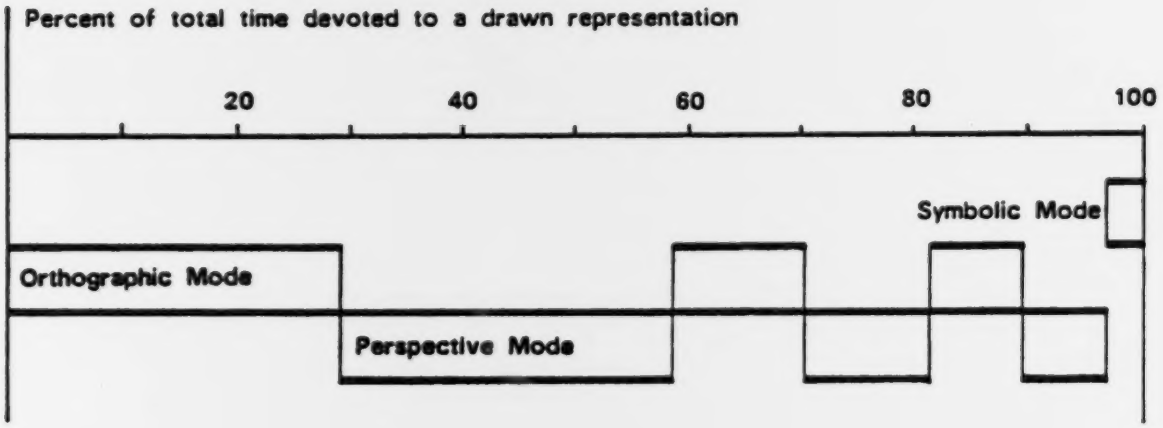
Coherence - Reconciling Information Between Sketches

A sketch often includes several views of an object, all of which should be in agreement; a common example is the two or three orthographic views of a form and the perspective sketch that frequently accompanies them. We have some interesting protocol evidence of a subject dealing with the coherence of a sketch. In reconciling between orthographic and perspective views two patterns were observed.

First, as the subject alternated between the views - between orthographic and perspective - the amount of time spent on each view decreased in a very regular sequence, less time with each alternation; starting with 15 or 20 minutes on each view and going down to a minute or less per view.

Figure 11

Pattern of Switching Views



Second, the piece of information - a shadow, a detail line, etc. - that was the last to be included in one view was consistently the first to be included in the next view. For example, the subject is working on the perspective sketch, adding details that represent design decisions. The last detail he adds is a representation of light reflection that indicates the digital readout is covered with a transparent window. The first detail he adds upon returning to the orthographic sketches is the same representation of light reflection.

The information processing model of human problem solving indicates a very limited capacity to carry complex information in short term memory. These patterns of alternation and information carryover seem to be a way of minimizing the visual information that has to be processed at one time. The subject stays in one view and shifts from one

design subproblem to the next until the current view is insufficient for solving the next subproblem. Or he stays with one subproblem (representing the transparent window, in our example) and shifts to another view.

Precision - Consistency Within/Between Sketches

Choosing a level of precision is clearly related to issues of economy and available time; a rough sketch is not simply the effect of sloppiness. If that were the case, there should be a randomness to the precision of linemaking within a sketch or between sketches. We have made comparative measurements of line precision that disproves this hypothesis. Rather, the precision of the lines within a sketch is consistently related to the length of those lines – longer lines are consistently less precise – and the precision of entire sketches changes systematically from low to high as the design processes moves toward completion.

Graphic output from our study included two sketches of one design solution; one a rough sketch produced at an early stage of the design process, the other a more refined sketch produced about 30 minutes later. Both sketches were similar in size. The refined sketch was somewhat higher in inclusion and coherence too, but the obvious noticeable difference was an increase in precision. Various straight line segments of the two sketches were chosen at random and compared for agreement with a geometrically straight line. An envelope was measured for each line segment; the envelope being a long rectangle with the line segment's length being the envelope length and the deviation from straightness being the envelope's width. A "perfectly straight" line would have an envelope width of zero. Wiggles in a line or multiple strokes in a line both contributed to lower precision and hence to a wider envelope, but the "weight" of the line did not count toward envelope width.

While the number of strokes comprising a single line varied from one to five, only single stroke lines existed in sufficient quantities to make reasonable comparisons. When length of line is plotted against deviation from straightness, the plot points that represent the lines in one sketch fall in a linear band indicating that deviation from straightness is a consistent function of line length. Further, the slope of the band is steeper for the rough sketch and has a higher zero intercept than the band for the refined sketch; an indication that the subject is controlling the precision of his sketches.

Precision - The Value Of Imprecision

Imprecision seems to have a value all its own. Our theoretical model says that in ill-structured construction tasks, invention (the substituting or adding of new ideas) goes on throughout the design process. By implication there is the prediction that we should see final solutions being postponed as long as possible.

In this context, low precision can be seen as compatible with the tendency to postpone final solutions, thus contributing to invention throughout the design process. At any stage of the design process a designer is more willing to abandon the ideas represented in economical, low precision sketches than those that represent a large commitment of time, thought, or money.

Finally, sketches that are low in inclusion, coherence, or precision can provide a dissonant visual field which encourages invention; perhaps toward the solution of problems that were not approached at the beginning of the sketch. The dissonances that occur, especially failures of coherence or inclusion, may be partly intentional and partly the result of errors in information carried over between views. However they occur, the designer periodically checks for agreement between views and discovers these dissonances. They tend to fall in areas of the sketch which have not been thoroughly solved; thus the dissonance can be interpreted as representing a range of possible solutions which require further refinement or perhaps the invoking of additional constraints or design standards. Alternatively, the dissonance can be interpreted as a spatial paradox or geometric impossibility which needs an instance of invention for its resolution.

Concluding Observations

Sketching is not so much the precursor of a drawing as it is of a thought. It is a way of controlling the information in the early representations of a design solution. The process can be applied to a wide range of representational modes.

In sketching, designers and engineers are engaging in a range of intentional acts.

1. They make an economic choice about what kind of representation and what level of inclusion, coherence and precision is most appropriate to explore the problem at hand.

2. They perform complex information processing by extracting information out of their own experience and out of previous sketches, adding information, and reconciling information conflicts in their sketches.

3. Through exclusion, incoherence, and imprecision they provide their sketches with enough ambiguity so that they can take advantage of inventive opportunities right up to the end of the design process.

It seems clear that the more kinds of representation a designer can use, the better able he or she is to work through complex problems.

Systems which force a designer to make an early decision about the inclusion, coherence or precision of information are counterproductive. They will tend to close down a designer's inventiveness too early in the design process.

Information Management

Introduction

A noticeable feature of the design process is the large amount of external information that the designer uses in reaching a final solution. For example, in the first 160 minutes of our in-depth study, the subject used a 9 page text, scale drawings of 10 components, foam-board models of 7 components, cost analyses of 4 partial solutions, photographs of 4 partial solutions, rough perspective drawings of 4 partial solutions, and partially rendered perspective drawings of 4 partial solutions.

The large amount of information present in a design task poses the designer the problem of how best to retrieve and store it. For example, after arranging foam-board models into configurations that represented partial solutions, the subject was interested in calculating the cost of each configuration. A possible technique for doing this involves looking up the cost of each component in the text. However, this requires fetching the text, matching each component with its written description, and then retrieving its cost. A more efficient technique – and the one used by our subject – is to write the cost of each component on its model beforehand, thus avoiding the necessity of searching the text. This technique reduces the designer's workload for retrieving cost information. In the following paragraphs we describe several information management techniques that the designer uses for reducing the workload of retrieving and storing information.

Information Management Techniques

We have identified 4 primary techniques that the subject used for managing information. The first, Partitioning, involves dividing stored information between different locations that vary in their accessibility. The second technique, Grouping, is creating groups of representations that are used in decision-making. The third technique, Labelling, identifies objects as belonging to specified sets. The fourth technique, Combining, is adding

information from one representation onto a different representation that does not already include it.

Partitioning The designer maintains two distinct areas for storing information which are ordered by the accessibility of the information contained in them[1]. The most accessible area is the subject's desktop and this contains the information and representations currently being worked on. The next most accessible area is the wall area around the desk on which the subject pinned various drawings of partial solutions.

The desktop contains most of the information that the designer uses for decision-making at any one time. Information needed for making a decision, but not present on the desktop, is typically brought here. For example , the subject would occasionally take a drawing from the wall and trace some aspect of it. Information on the walls is generally used for reference. For example, the subject would take a measurement from a drawing pinned to the wall, and use it to confirm some dimension of a drawing on the desktop.

The designer also has access to an enormous amount of information that is not kept on the desktop or on the walls, e.g. drawings from previous design projects, tables of standards etc. This type of information differs from the other two because it is not all contained in any single area. Furthermore, this information is not as accessible as that contained in either of the other areas because most of it is not immediately visible.

Partitioning is useful as a management technique because separating information into separate areas reduces the workload of retrieving information from any single area. Furthermore, the ordering of the areas (in terms of their accessibility) ensures that there will be minimal interference between the information contained in the different areas.

Grouping The two characteristics of this technique are that sets of information are grouped together, and the groups are defined by virtue of having different locations. This technique is similar to Partitioning. except that the groups are only temporarily assigned to locations. In contrast, the areas used in Partitioning are unchanged throughout the design process.

Probably the simplest and most common example of grouping was the subject's tendency to create triads of drawings on a single sheet of paper: a plan view, a side view and a perspective view. In this way, the subject groups several sets of information – the different views, at a single location – the sheet of paper. Another example occurred at the

[1] The designer also maintains areas for manufacturing objects: a small woodworking shop, and a desk for light materials such as paper. These areas are excluded from Partitioning because they are concerned with manufacturing rather than storing and retrieving information.

beginning of one session when the subject placed all the photographs, models and rough sketches into separate groups around his desktop while he was reacquaintancing himself with the design problem.

The subject also used some more complex versions of the simple Grouping technique illustrated above. In one example, the subject explicitly used the order of the groups: After completing all the foam-board models, he created a 'matrix' of the models with the different types (e.g. keypad, credit-card reader) arranged horizontally and the different versions of each type (keypad #1, keypad #2) arranged vertically. This matrix enabled the subject to rapidly retreive each choice of model. Furthermore, by retrieving and replacing each type of model from left to right, he knew that he had considered every type when he replaced the rightmost model. Thus the matrix version of Grouping appears to provide the designer with an inexpensive way of sorting through a collection of objects.

Labelling This is simply the act of giving an identifying label or name to a partial solution. The label identifies an object or a collection of objects as being part of a (labelled) solution. The subject maintained labels for each of four partial solutions that he generated. These labels were transferred as each new representation of a partial solution was generated, for example: from photograph to rough perspective to rendered perspective. The labels on the photographs were numbers written on pieces of paper which were glued onto the photographs, and the labels on the perspectives were simply numbers written on the page.

During one session the subject made an error which illustrates the importance of maintaining a labelling scheme. He began to make a perspective drawing which he had already completed, indeed he used the original perspective to help format the new one on the page. When the experimenter asked the subject which partial solution he was working on, he first compared the labels of the new and old perspectives and wondered if he had mislabelled one of them. However, after he had compared the drawings themselves he realized his mistake. This example suggests that the labels provide the designer with a quick method of distinguishing different solutions instead of comparing details in the drawings, which is a relatively slow process.

Combining This type of activity was the one alluded to in the introductory paragraphs. Specifically, the subject would append information to a representation which did not include that information. For example, the subject always wrote the price of a component onto its foam-board model. Thus the cost information – something that is not inherently represented in a foam-board model, was appended to the representations of the components.

In other instances of Combining, the subject wrote the cost of each component in a configuration of models and glued it to the photograph of that configuration, and he annotated the rough perspective sketches with approximate dimensions.

Combining information on the various representations appears to free the designer from unnecessary searching. The subject added information which was not inherent in a representation, but which was useful for making decisions which involved considering information from different representations. One implication of this technique is that the designer already has a good idea of what information will be useful in making decisions.

Conclusions

The designer needs to consider a large amount of information in order to create a good design. However, this requirement also means that the designer needs some techniques for organizing or managing this information. These techniques help to make the design process more efficient by minimizing the amount of time spent on retrieving, searching, and sorting information. Additionally, these techniques can also minimize errors such as losing or regenerating information.

We observed the subject using four information management techniques: Partitioning, Grouping, Labelling, and Combining. Two of these techniques rely heavily on a spatial cue - location - for organizing information. The other two techniques involve embellishing the contents of a representation, and maintaining a labelling scheme.

User Documentation - Module Two

Module Two Research and Implications

Background

Module One of the CMU CADAM Manual was completed in December 1982 and was well received by IBM. This prototype manual was found to meet the needs of novice CADAM users by providing clearly defined goals, itemized procedures, and simplified reference aids. The success of the manual was primarily due to the use of Protocol-Aided Revision (PAR)[2] to test and revise the manual, and the diverse team of experts who contributed to this manual.

Module Two, just completed, was supported jointly by the Kingston Design Center and the MDA Education Task Force. The goal of Module Two was to develop a comprehensive basic training manual using the same research strategies implemented with Module One. Module Two is the second part of a two module set. Two support documents, the Reference Summaries and the Pocket Reference Guide, were written to accompany the set.

As a result of our work on Module Two we have identified several concepts that should influence future development of Computer-Aided Design software manuals and computer manuals in general. Some accomplishments are:

1. Accommodating a large range of users by using integrated layering of novice and expert information needs. In general, gaining access to information about the software has been greatly improved over previous CADAM manuals.

2. Reducing the time and cost required to learn and become proficient with the software.

3. Using visual cues such as symbols and graphic layout to assist the user by reducing the reading load and by presenting the material in easily managed chunks.

4. Providing clear, concise and organized verbal information.

This report summarizes our research findings during the development of Module Two. For more information on the draft development of Module Two see Appendix A.

---

[2] For more information on the PAR methodology and the research used to write and design Module One, see CMU/IBM Usability Study, Interim Report-Number 2, p. 25, 1982.

Information Density

When we examined existing CADAM manuals we were disturbed by their size and the information density of the page layout. For example, an average instruction page in an existing manual has 200 to 250 words, no diagrams, and few reference aids. An average instruction page in the CMU manual contains 75 to 100 words, has one or two diagrams, and includes reference aids such as page headings and numbered actions.

Some manuals we examined were 400 to 600 pages of densely packed information. Our first drafts of Module Two were about 250 pages and the Reference Summaries were about 150 pages. The initial drafts included a minimum of explanatory information. However, the protocols reveal that novices need more information and even expert computer users need descriptive information about how the software works. Our final draft of Module Two is 478 pages and the Reference Summaries are 248 pages.

Although our manual is larger than expected, it has three characteristics that make it different from other large manuals.

1. The written information is much more concise and clearly written than other manuals, and the information is organized in a way that decreases information search time and increases comprehension.

2. White space (page areas left blank) functions as a graphic design element as well as an information retrieval aid. By not crowding the page and by following a specific verbal and graphic organization, the manual's usability is greatly increased.

3. A major percentage of the page space is occupied by diagrams, which few other manuals had. These diagrams proved to be one of the most significant contributions to the manual in terms of helping the user to understand complex software concepts and to guide the user through the various lessons.

The real issue then, is, not how many pages a manual has, but how inviting the pages look, how well they communicate, and how much time it takes the user to work successfully through the tutorials.

Defining the Audience and Structuring the Manual

The entrance of computers into professional fields brings inexperienced people in contact with some unfriendly software. For example, although CADAM was designed to do technical drawings and numerical control applications, secretaries are using it to prepare presentation overheads.

The protocols of Module Two reveal that a user's level of computer experience makes little difference in the amount of information about CADAM needed to learn the system. The "computer experts" need the same information as the novice users.

This result suggested that we needed to redefine our understanding of the audience for this manual. Originally we categorized the audience as computer experts, technical (subject matter) experts, and novices. As the manual progressed, we realized that 1) everyone started out as a CADAM novice and 2) after working through the manual once, users would need ways to find answers to questions, reguardless of how often and for what reason they would continue to use CADAM.

The notion of "layering" is a solution to the problem of user audience. Layering is usually interpreted as writing three different manuals: one for the novice, one for the occasional user, and a technical reference for the expert. This approach is not time or cost effective and will most likely deprive some users of needed information.

The unique organization of Module Two and its accompanying support documents are examples of integrating layered information. Module Two accommodates the novice by providing lessons, function overviews and sufficient explanation. Novices also made extensive use of the Pocket Reference Guide and the Reference Summaries during and after the protocol sessions. Expert strategies were incorporated into the lessons and included in a separate section so that the novice users could learn and access these time saving techniques. The occasional user will benefit from the Pocket Reference Guide, the Reference Summaries and, from time to time, the tutorials. The expert user will benefit from the Reference Summaries which support the more technical manuals.

Time

We recommend that the manual be used as follows: work through a chapter and then create a practice drawing. Without practice it is difficult to retain the information for long periods of time. In addition, using the manual like this breaks up the learning process into manageable chunks.

Our goal in developing Module Two was to create a self study manual that would take 40 hours or less to work through (including Module One). The actual time taken to complete Module One is 2 to 3 hours. The actual time taken to work through Module Two is 9 to 11 hours. The total time to work though both modules is 11 to 14 hours. These times do not include time taken for coffee breaks, lunch, or practice sessions. When all the breaks and practice sessions are included, it takes the average user about 3 or 4 days to complete the manual.

The following is a comparison of the time, expense, and support previously required to become proficient at using CADAM and the results obtained by our protocol subjects.[3]

Table 1

Time Comparison

	Existing Training Method:	CMU CADAM Manuals:
Time to learn the software basics:	One week of study with an instructor.	3-4 days self study.
Time to become proficient with the software:	2-3 months . of hands on experience.	2-3 weeks of hands on experience. (observed protocol subjects)
Time to become expert:	One year of hands on experience.	We *estimate* 6 mos. using the reference summaries. (based on observation)

Cost Comparison

Manual Source	Manual Cost	Student Overhead	Instructor/ Tuition Cost	Total
CADAM Inc.	$600	$1,600	$1,560	$3,760
IBM Internal	$10	$2,000	$100	$2,110
CMU	$10	$1,050	$0	$1,060

Student overhead is calculated at $50 and hour. Instructor cost is calculated at $50 an hour. Data for the cost of CADAM Inc. manuals and instructors was supplied by CADAM Inc. and was dated April 1983.

[3] Data for existing manuals provided by IBM Kingston

Information Access

One of the most significant advantages of the CMU CADAM Manual is that it provides several ways to access information. This is important because users quickly advance from being novices, who need step by step instructions and explanations, to either occasional or expert users, who need to find answers to specific problems quickly. Access to information is provided by:

- Detailed Tables of Contents
- Descriptive page headings
- A hierarchy of headings within the text
- A Table of Contents for each section of the manual
- An Index
- A hierarchy of information based on the overall organizational plan and reflected in the graphic design and page layout.

The development of pageheadings is a good example of how PAR can help improve a manual. Pageheadings in the initial drafts included the page number and chapter title. However, the protocols revealed that subjects had trouble finding information because the pageheadings were too general. We therefore revised the pageheadings to include both function name and the specific procedure being presented.

General Organization and Concept

The tutorials were structured as drawing lessons in which the user creates a drawing of an object. Within each drawing lesson, simple exercises helped the users to learn the software basics. In addition, the drawings increased motivation because they were different enough from the usual, everyday drafting problems.

Module Two is organized as follows:

Each chapter in Module Two is a tutorial designed to gradually expand the skills of a prospective CADAM user. Each chapter has the following sections:

Previews	describe the basic goals of each chapter and include step-by-step diagrams of the drawings to be constructed.
Start-ups	give basic instructions about how to get started.
Exercises	include the following types of information:
	Overviews describe overall capabilities of a function or the procedures associated with a particular menu selection and, in most cases, include diagrams to illustrate a procedure or concept.
	Action Series are brief descriptions of what will be done in the next series of Actions, and why.
	Actions and *Responses* give detailed instructions on how to carry out a procedure on the CADAM system.
Chapter Summaries	list the procedures and strategies learned in each chapter and encourage users to practice what they have learned.

After the first set of protocols we incorporated needed information in the form of Tips, which appeared one line below the related Action or Response. The type of information in a Tip ranged from "Watch the message area" to "The system will not respond immediately." However, subsequent protocols revealed that subjects would read the Action, carry out the instruction, and miss the Tips entirely. To solve this problem, we incorporated all the necessary information into the Action. For example: "Select the point at 100,100 and watch the message area for confirmation."

Symbols

Our success in Module One with using symbols for "select" and "indicate" encouraged us to incorporate more symbols in Module Two. For example, originally the menu or message area was represented in a space about 4 inches wide with all the screen data appearing in

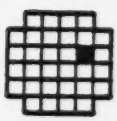
a representation of the menu area or message area. However, after the first few protocols we realized the subjects were still not reading the message area or menu area. Although the response was included in the manual, it was not signaling the user to learn to look at the screen area. To solve this problem we developed a small symbol about an inch long to represent the menu and message areas (Figure 12). The men or message area information that the user needed to see was typed under the symbol. This signaled the user to check the message area and match it to the response typed in the manual. The symbol worked very well, training the users to watch the message area as they worked.

Figure 12

Message Area symbol

Menu Area symbol

Menu/ Message Area symbols shown in the context of the manual



A ▼ the DIMENSION function key.

DIMEN VIEW PV SCL 1.000 WDO 1.000,.0
SEL PT / SEL LN / YN DATUM

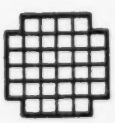
/ HZ / VT / PR / AN / NM / RD / DM / CV / CH / SHO / N-S /

We further noticed that the "select" and "indicate" symbols tended to occur at the beginning of each Action. After looking at all of the Actions, we also noticed that the words "Hit" (for hit the function key) and "Key in" (for key in something on the alphanumeric keyboard) occurred at the beginning of many Actions. To decrease the reading load on the user, we designed symbols for these words that would work similarly to "select" and "indicate". These symbols were also successful.

In Module One we used the symbol in conjunction with the word, for example: "Select ⊙ the point at 0,0." In Module Two we used this same presentation for the first lesson and then we dropped the use of the word in following lessons, for example: "⊙ the point at 0,0." Most subjects did not notice the words missing; subjects who did asked why we didn't do it sooner.

Figure 13

Symbols currently in use

Symbol	Meaning
A	Action
R	Response
⊙	Select ⊙ with the light pen
◍	Indicate ◍ with the light pen
▽	Key ▽ in information on the CADAM® alphanumeric keyboard
▼	Hit ▼ a function key
	Represents the function keyboard; included to help you locate specific function keys

Diagrams contributed to the manual by reinforcing the Actions and Responses. They also make the manual appear user friendly and less complicated.

Conclusions

We believe that our manual is successful for two reasons:

- We used PAR to test the manuals.
- The manual was produced by a team of experts in different fields.

Debriefing interviews of subjects revealed that subjects forgot problems they experienced in using the manual and the system. The protocol tapes captured the actual experience of the users and gave us valuable information for improving the manual.

The composition of our team was another major reason for the manual's usability. One problem with other manuals is the tendency for programmers to write manuals for programmers. Our team included writers, designers, technical experts and real users. We feel that all manuals should be written with this combination of skills.

An additional benefit of our manual is that it can be produced easily. The text is edited and formatted using EMACS and SCRIBE. The symbols that appear in the text are special characters designed specially for this application and are easily inserted into the text with one command. The manual is one color (black ink) and can be printed or photocopied. All of the drawing diagrams and the menu and message area diagrams were produced using CADAM and plotted using the IBM 7374 plotter.

User Aids - Computer-Aided Instruction

CADAM/CAI

Introduction: Computer-aided instruction compared to printed documentation

The goal for this deliverable was to develop CAI/CADAM, a computer-aided instruction module for CADAM software, that was comparable to Module One of the existing CMU CADAM printed manual. These modules could then be tested to compare the two presentation modes; computer-aided instruction vs. printed instruction. We also included a comparative pilot test of the two modes. It has implications for future investigations of CAI presentation.

CAI/CADAM is designed with respect to cognitive and design principles developed at CMU. It maintains the same approach to CADAM instruction as the printed version of Module One except where the CAI medium makes reorganization necessary, or improvements possible.

Details of CAI/CADAM's design, the pilot test comparison and concluding observations are included in the following report and in Appendix C.

The major implications of this investigation are:

1. The overall speed and completeness of learning CADAM by CAI seem to be more affected by attitudes and knowledge that the user brings to the learning situation than by differences in presentation.

2. Tutorials should address drawing strategy when using CADAM as well as the procedures. For both CAI and printed instructions, the major nonproductive delays were caused by gaps in the subject's understanding of the strategy that CADAM requires to draw any drafting.

3. CAI presentation seems to give users a better understanding of the overall sequence of procedures required to use CADAM. There were fewer false starts and errors, especially in the information management procedures.

4. If the screen retrieval rate for CAI were faster, there might be an improvement in learning times over written presentation. During early learning in either mode there is frequent referencing. CAI seems to streamline the access to references but the slow screen retrieval rate apparently cancels the advantage.

Design Principles

Consistent with the design of CADAM printed tutorials, a major emphasis of our CAI design was usability. The CAI module uses the same principles of of human cognition, graphic design and writing that had previously been developed at CMU and relies on research done by Dr. Kamila Robertson regarding user interaction with information presented on CRT screens.

Because comparison of CAI to printed presentation was the major goal, the CAI design intentionally parallels the large scale organization of the CMU CADAM tutorial Module One. That is, a short introduction, followed by a series of sequential exercises including reference summaries at the end of each exercise, then a glossary and a help section. The CAI medium, however, presents redesign opportunities for the presentation of information on screens and the access to that information via menus, scrolling, and contextual locators. Particular solutions are described in more detail in the following sections.

The principles of individual screen design were derived from Robertson and from our experience with printed tutorials; they include:

- Displays must be clear, concise, and consistent in both writing and graphic design.
- A limited set of display formats should be devised in response to both content and user needs, and then adhered to.
- Text screens should be limited to eight lines or less.
- The most important information should be located toward the top of the screen.
- When lost, expert users search for information by scanning the screen perimeter clockwise; novices stare at the center.
- Users of CAI interact with it more effectively when they have a conceptual map of the whole system.
- Contextual locators are useful in building and traversing the map, but are confusing if overused.
- A user can deal with only two levels of hierarchy at a time: The current one and one above or below.
- Novices are very attentive to minute discrepancies.
- Evaluation and editing by Protocol-Aided-Revision (PAR - See Appendix A for description) is part of the design process.

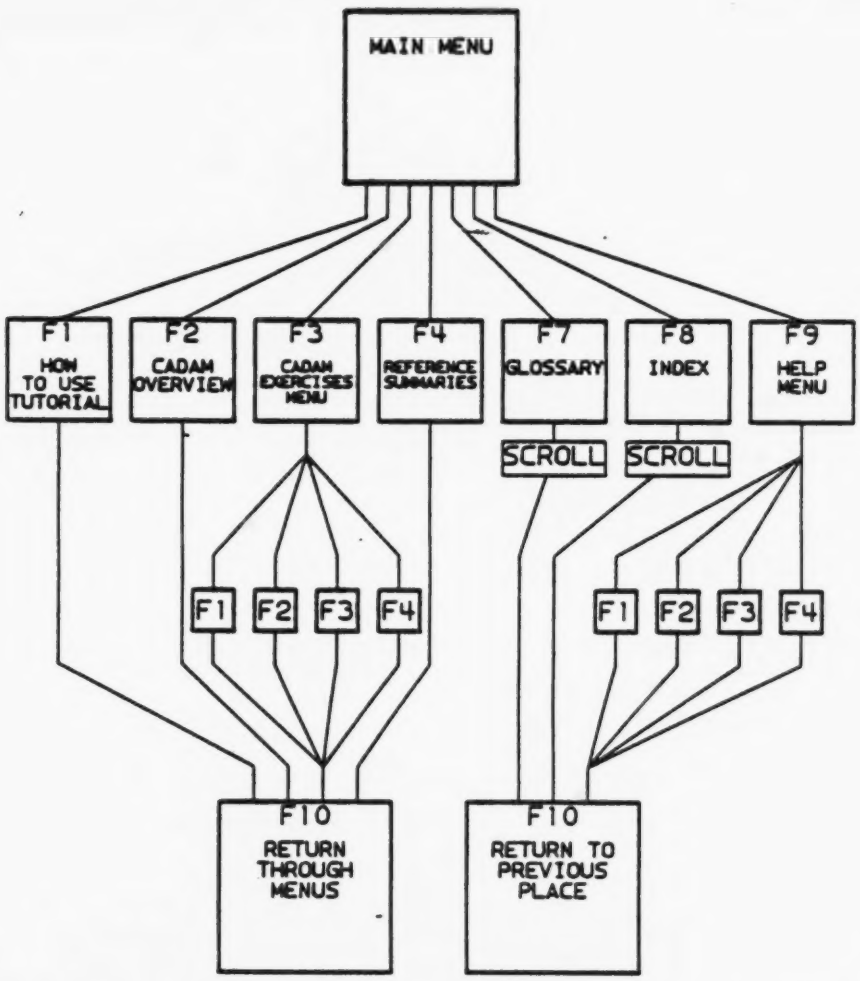
The Conceptual Map and Information Access

To aid in the users understanding of the module we made its structure as simple and obvious as possible. This helped the user form a conceptual map of the whole system. Once this conceptual map was formed the user could "travel" through the module without getting "lost".

Users can only deal effectively with two levels of hierarchy at a time, the current and the last or the current and the next. By providing cues on each screen to the last and next levels we were able to aid the user in identifying his place in the tutorial.

Figure 14

Information Access Diagram



Since the module is used as a tutorial and a reference document, it is necessary to access its information from several points and along several paths. In addition to movement forward and back through the module, there are branching points which also permit looping; push down/pop-up access to the glossary, index and help; and clear-path returns through the information network. See figure 14. All these actions are controlled by P.C. function keys:

- Backward/forward This is the basic linear motion throughout the tutorial. It is always controlled by the F1 and F2 function keys;
- Branching menus The main menu is the principal control of paths through the module. In all menu screens, the graphic layout of menu choices corresponds to the layout of the function keys which activate the choices. Other menus provide a choice of exercises and several kinds of aid;
- Push-down/pop-up Glossary, index, and help are accessible at any point in the exercises. Glossary and index are accessed through an alphabetically ordered scroll. Help is accessed through a menu which sends the user to the appropriate place in the module. The current place in the tutorial is marked while the user pushes down to any of these aids; when finished using the aid, the user is popped up to his previous place in the tutorial;
- Clear-path return At all times the F10 function key is reserved as a means of returning through the hierarchy to the previous choice point; it is a master pop-up control. A series of F10 actions will always eventually allow the user to exit the entire module.

Operating the CAI environment

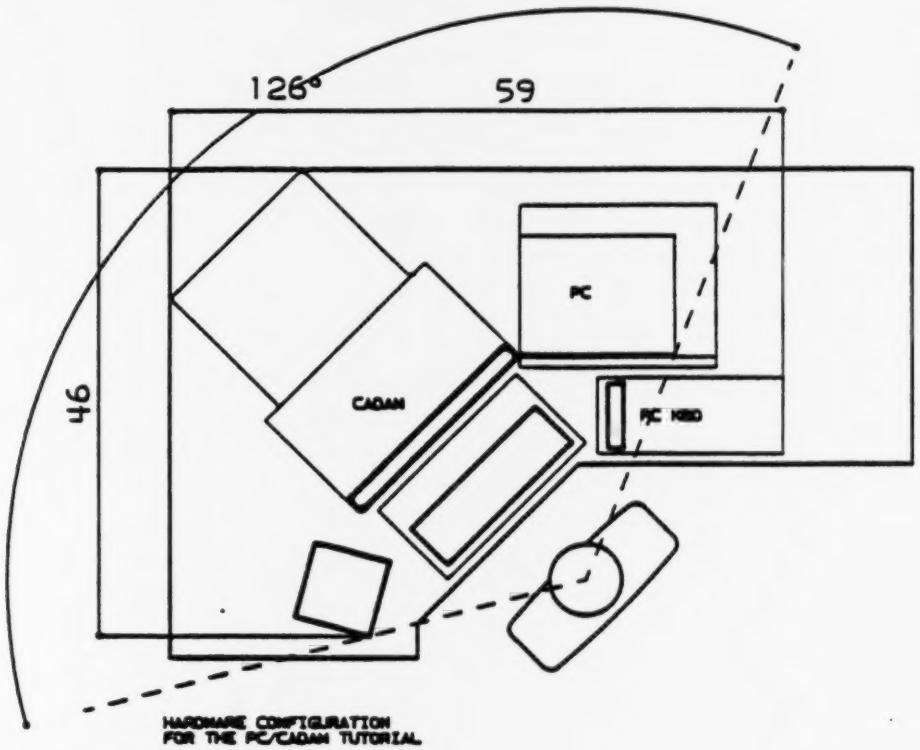
CAI/CADAM runs on a PC/XT. One of the disadvantages that CAI has over printed manuals is the overwhelming amount of equipment the novice computer user must face to learn one software package. It became clear by analyzing the video protocols that the appropriate arrangement and operation of this equipment is not obvious. It must be learned along with the contents of CADAM and its implied drawing strategy.

To minimize this environmental component of the learning, we restricted the amount of interaction the user had with the PC. The PC essentially runs itself, the only keys the user has to touch are the ten function keys located on the left hand side of the P.C. keyboard. The pertinent keys are labeled on each screen. This proves to be quite effective with the user learning the meanings of the function keys quickly. Since many novice computer users do not have keyboard experience, minimal interaction is important.

In addition to the possible complexities of operation, the CAI physical environment can easily become cluttered. One of the problems we faced in developing the CAI module was the ergonomics of the workstation itself. Problems included the size of the PC as compared to available space in a CADAM workstation, accommodating lefthanded users, and the physical relationship of the PC screen to the CADAM screen.

Figure 15

A plan view of the CADAM workstation with the PC/XT equipment



Since the P.C. operation depends only on its function keys, it was not critical that the PC keyboard be located directly in front of the user. This is especially true for F1, F2, and F10 which are at the corners of the key area and get most of the action. However the keyboard and screen had to be within easy viewing distance and reach of the user. A right handed layout is shown. The left handed layout is more difficult. The PC and CADAM function keyboard are interchanged and the PC keyboard is rotated 90 degrees to the user between the two screens.

Screen Formats

To graphically accommodate the variety of information and access to it, we established four basic screen formats – Instruction format, Menu format, Scroll format, and Reference format. Of equal significance, we required that every screen adhere to its appropriate format. In all the formats, two areas remained consistent – a header and a footer. The area of the screen between the header and footer was for active information and changed in response to the information being presented. This included menus, text, exercise actions and responses, scrolling locater and summary actions.

The header contains the following elements:

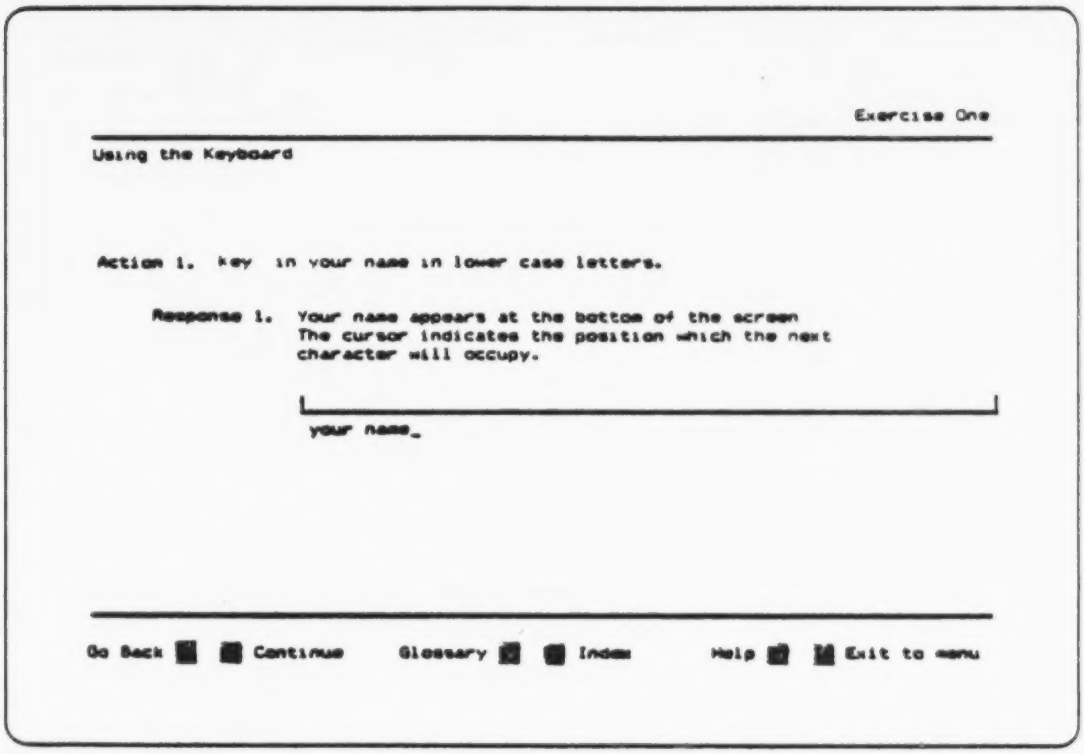
- A rule running across the top of the screen, to locate the header area as well as to define the area below as the active information area.
- Above the rule on the right is the section locater. This is the name of the topic, such as "How to Use This Tutorial", or the name of the exercise, such as "Exercise Two". When the user was in the HELP or INDEX function this header area was represented in reverse video and the word HELP or INDEX also appeared.
- Below the rule on the left is a heading that identified the particular information being presented on that screen.

The footer contains the following elements:

- A rule to define the active area of the screen and to locate the footer area.
- A signal for continuing the tutorial is included just above the rule on the right.
- Function key prompts are located below the rule and indicate what each function key does. F1 and F2 always retain their backward/forward meaning, F7, F8 and F9 are always Glossary, Index and Help, and F10 always retains its pop–up meaning although the specific prompt will change to be specific to that section of the module, e.g. "Return to main menu" or "Return to your place in the tutorial". Function keys which may be disabled in a particular section of the module are not shown in that footer.

Figure 16

Instruction Format: Action/Response

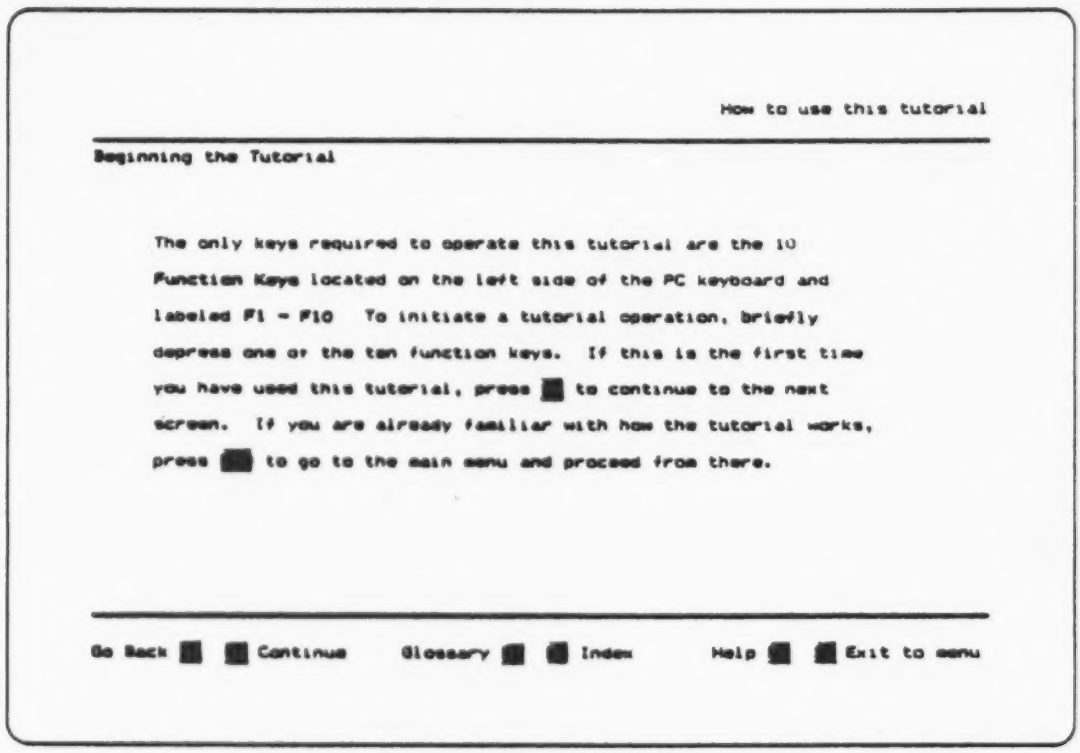


This example of an instruction screen (fig. 16) an action/response sequence in the active area. In exercise one, the entire words "Action" and "Response" were used instead of "A" and "R". This example also shows one of many diagrams – a representation of the CADAM Function keyboard with the appropriate key blinking.

For readability, action and response lines were kept to an average length of 4 inches and a maximum of 6 inches. On text screens the line length averaged 5 to 6 inches (50 to 60 characters) with a maximum of 7 inches. The text screens never contained more than 8 lines of text and were double spaced. Actions and responses were single spaced but had ample spacing around them, and their associated diagrams.

Figure 17

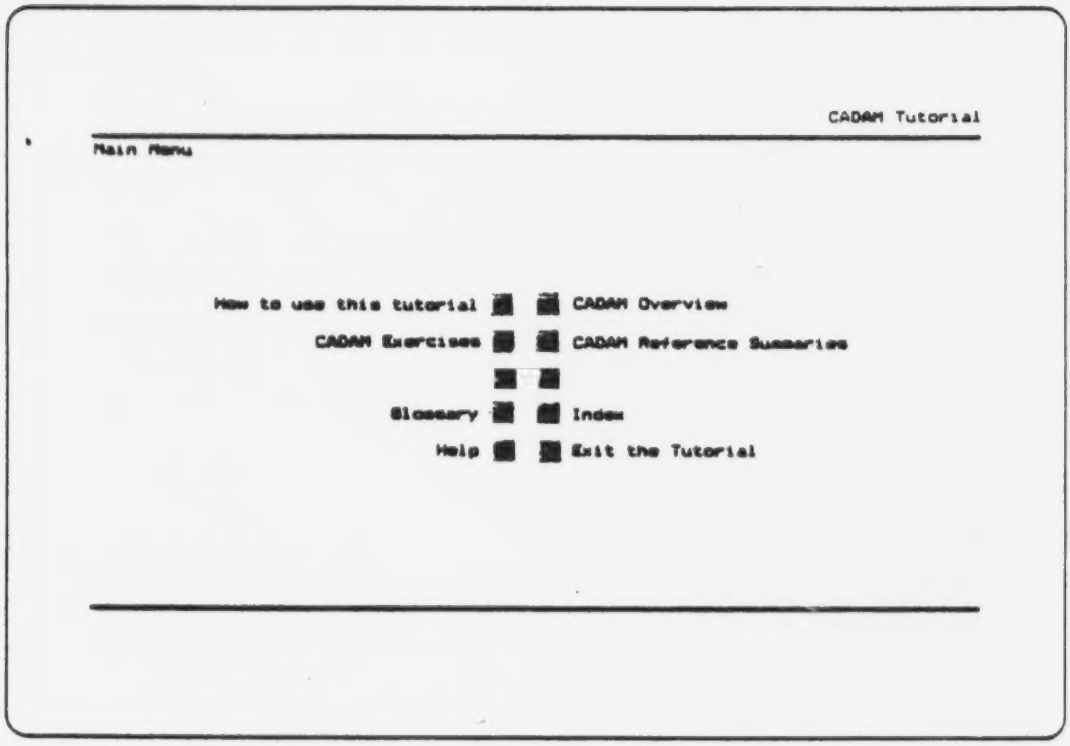
Instruction Format: Text


How to use this tutorial
Beginning the Tutorial
The only keys required to operate this tutorial are the 10
Function Keys located on the left side of the PC keyboard and
labeled F1 - F10 To initiate a tutorial operation, briefly
depress one of the ten function keys. If this is the first time
you have used this tutorial, press to continue to the next
screen. If you are already familiar with how the tutorial works,
press to go to the main menu and proceed from there.
Go Back Continue Glossary Index Help Exit to menu

This example (Fig. 17) shows the instruction format used as a text screen. It is fairly typical having seven lines of text ranging between 58 and 65 characters long.

Figure 18

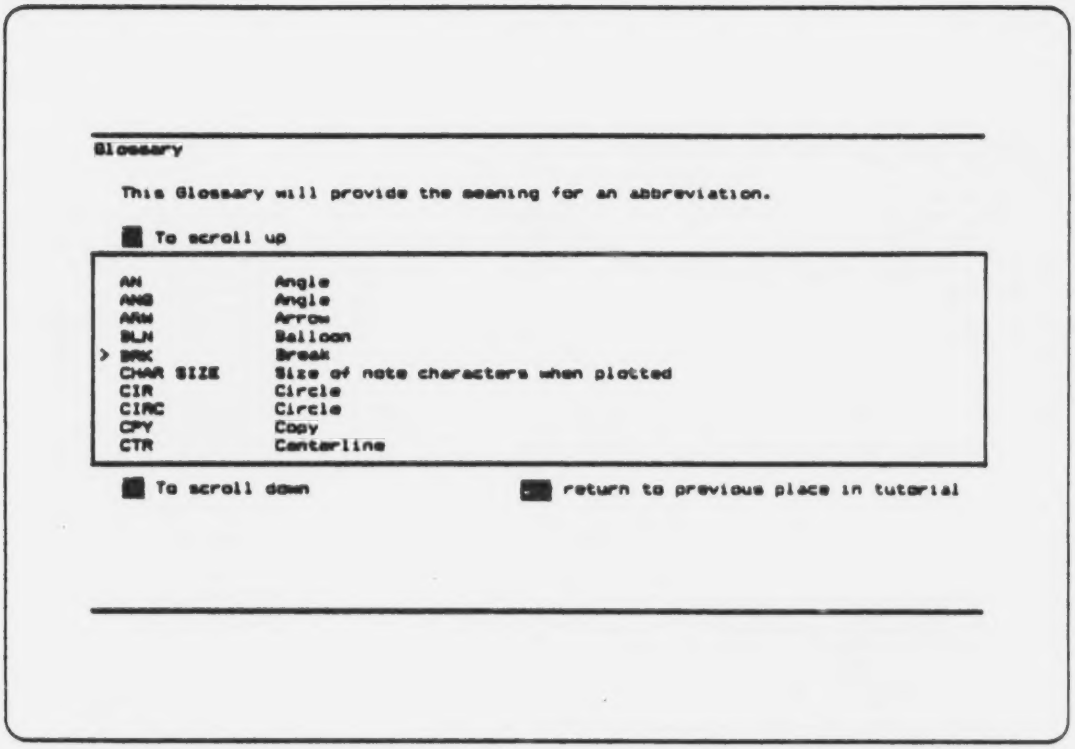
Menu Format



All menu screens retain the centered graphic layout shown above (Fig 18). The representations of the keys repeat the layout on the P.C. keyboard. In this particular menu F5 and F6 are not required and are left blank. On menu screens there are no prompts in the footer because all choices are represented in the menu.

Figure 19

Scroll Format: Glossary

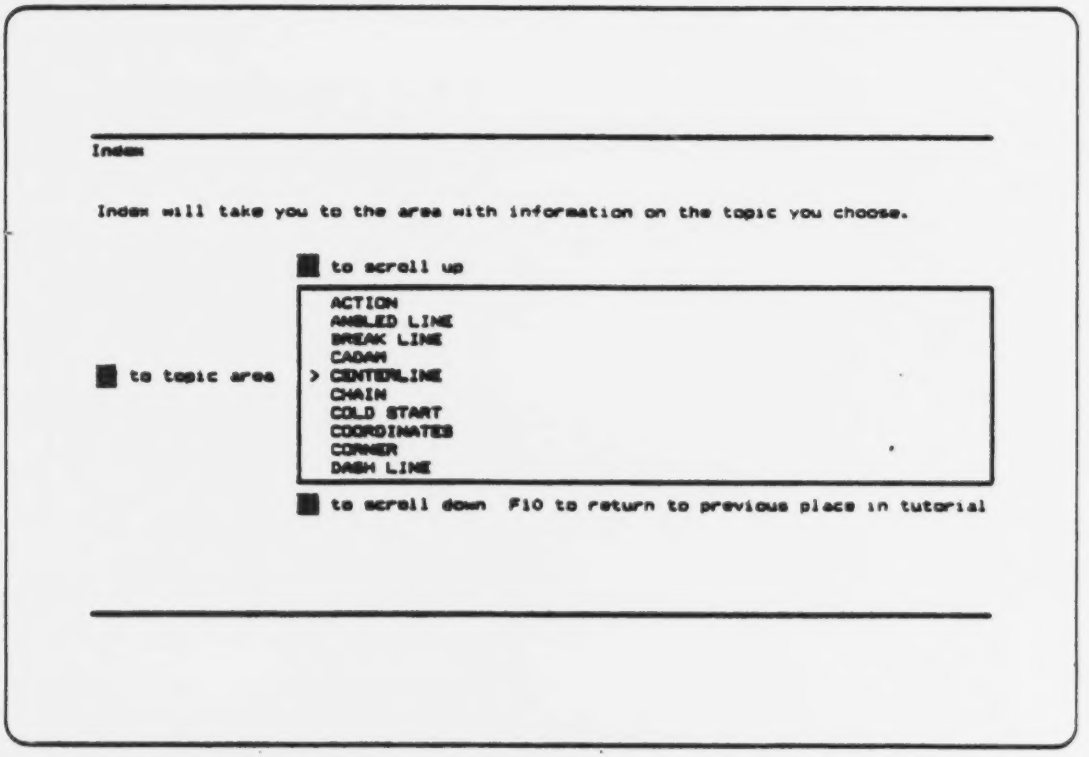


Scroll screens such as the glossary (Fig. 19) provide push-down/pop-up access to information and don't have operative function key prompts in the footer. F10 ends the use of the Glossary. The words inside the box scroll up and down. The target word and its meaning, next to the pointer, are displayed in high intensity.

The scroll screens are the only screens in CAI/CADAM that use a box to isolate information. With the CRT screen providing its own set of concentric visual boxes, additional graphic boxes usually result in unnecessary visual complexity. In the case of the scroll screens the box safeguards against unwanted visual illusions by separating the moving text from the nearby stationary text.

Figure 20

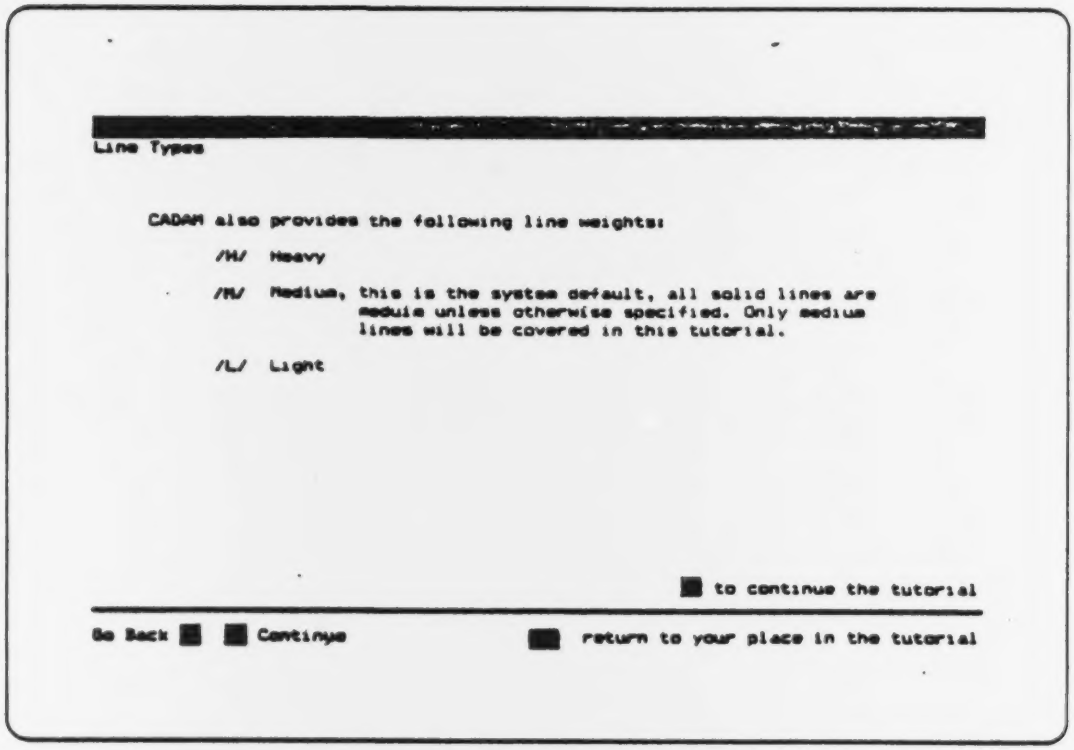
Scroll Format: Index



The index screen (Fig. 20) is similar to the glossary except that it leads, via F9, to a specific place in the exercises. Exercise screens appear; F1 and F2 allow the user to move back and forth through the exercise; then F10 pops-up the user to his previous place in the tutorial.

Figure 21

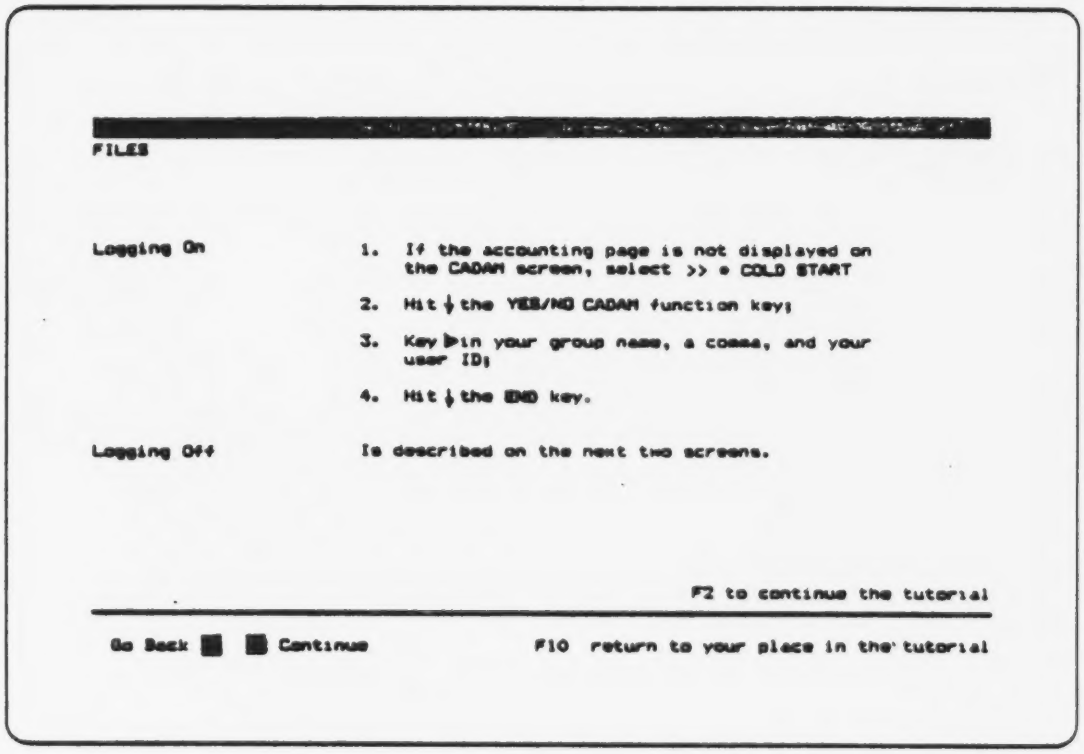
Reference Format: Index


Line Types
CADAM also provides the following line weights:
/H/ Heavy
/M/ Medium, this is the system default, all solid lines are
medium unless otherwise specified. Only medium
lines will be covered in this tutorial.
/L/ Light
to continue the tutorial
Go Back
Continue
return to your place in the tutorial

Reference screens are in a format that is similar to instruction format with a different header and footer (Fig. 21). Since the two formats are superficially similar, the heavy reverse video band on the header reminds the user that he is in a reference section and has to return to the main tutorial in order to continue. Further, all function keys except F1, F2 and F10 are disabled to prevent the user from getting lost in nested hierarchies such as might happen if one tried to enter the help section directly from the index section. Once popped-up to the main tutorial, all reference aids or menus are again accessible.

Figure 22

Reference Format: Help


FILES
Logging On
1. If the accounting page is not displayed on the CADAM screen, select >> e COLD START
2. Hit ↓ the YES/NO CADAM function key;
3. Key ▶in your group name, a comma, and your user ID;
4. Hit ↓ the END key.
Logging Off
Is described on the next two screens.
F2 to continue the tutorial
Go Back
Continue
F10 return to your place in the tutorial

The help screens are similar to the index screens except for the active information area. As shown above (Fig. 22) the active information is identical to the reference summaries that follow each lesson. These are written as a simple action series rather than the action/response series of the instruction screens. When reference summaries are accessed directly following an exercise or directly through the menu structure, the usual instructional header and footer appear rather than the reference header and footer shown above.

Graphic Cues

In addition to designing a grid for the screens we incorporated many graphic cues for the user. The use of highlighted or bright text on the screen worked well for signifying particular CADAM menu selections or important words. The highlighted words Action and Response (later A and R), helped to direct the user's eyes when they turned back to the PC after working on CADAM.

In modules One and Two of the printed CADAM tutorial we incorporated many symbols to signify action words. These symbols worked well in the printed tutorial because we were able to design and test them. We tried, unsuccessfully, to simulate the same symbols in the CAI module using the available keyboard characters. These characters turned out to be distant approximations of our printed ones (Table 2) and were a source of some confusion.

Table 2

Comparison of Printed vs. CAI Symbols

Printed Symbol	CAI Symbol	Meaning
A	Same	Action
R	Same	Response
⊙	>>	Select ⊙ with the light pen
◍	<>	Indicate ◍ with the light pen
▽	>	Key ▽ in information on the CADAM alphanumeric keyboard
▼	↓	Hit ▼ a function key

Experimental Design

CAI/CADAM was tested using Protocol-Aided-Revision, PAR to debug the program, revise the text and refine the general organization. The module was pre-tested on four subjects in groups of two. After each group we debugged the program and revised the text and screens.

For the comparative pilot test we recruited 12 subjects to participate in the experiment. All were design students with similar formal exposure to computing. The experiment was designed as follows. (See Table 3).

Two equivalent tutorial documents were created, one a version of our printed Module One and the other a computer-aided instruction diskette, CAI/CADAM.

Two equivalent test problems were designed to test learning immediately after the tutorial session and again, one week later.

The subjects were preassigned the type of tutorial, either printed or CAI, and to a test problem.

Table 3

Design of Comparative Pilot Test

Tutorial Type	No. of Subjects	Test After Tutorial	Test One Week Later
CAI	3	Test 1	Test 2
	3	Test 2	Test 1
Printed	3	Test 1	Test 2
	3	Test 2	Test 1

The subjects were allowed to take as much time as they needed to complete the tutorial. They were allowed to ask for help. Answers to questions were restricted to where they could find the information in their tutorial, thus no additional information was given.

When they completed the tutorial they were given a short break (about 5 minutes) then given the test problem and one hour in which to complete it. The subjects were not allowed to ask for help during the test, but they did have access to whichever tutorial they used in the training session.

Due to CADAM system problems (subjects unintentionally logging off the software job) only four clean, complete runs were possible with the CAI tutorial and only five with the printed tutorial. Under these conditions, the retesting one week later would have been seriously flawed so it was not carried out. Despite the small number of subjects, some patterns appear and are discussed in the results section. If there is continued interest in CAI tutorial design, a larger comparative test should be run with contingencies for system problems that are caused by inexperienced users.

Experimental Results

At a general level of comparison, there doesn't seem to be a significant difference between CAI presentation of tutorial information and the traditional written presentation. Some interesting patterns of difference begin to show if one looks at the time segmentation of actions that the test subjects performed when they were being tested right after their first encounter with the two tutorials.

Similarities

In both modes the total time spent on the tutorials varied between approximately 1 1/2 and 3 hours. (See Fig. 23) The average of all times was 2.2 hours for CAI and 2.1 hours for written. If one flawed run is eliminated from the CAI data, the average changes to 2.3. Including only those subjects who completed the test, the averages are 1.7 for CAI and 2.0 for written.

The scores from the two modes are again similar, 77% for all CAI and 80% for all written. Removing the flawed run changes the CAI average to 81%. Including only those who completed the test, the average scores are 92% for both groups. Since there was a one hour time limit on the test, the time spent on the test is only meaningful for those who completed it. Those who could not complete it in that time are grouped together as "incomplete in one hour". The average test time was .6 hours for the CAI group and .7 hours for the written.

Figure 23

Performance on Comparative Pilot Tests

CAI/ CADAM

	Time to complete the Tutorial	Time to complete test	Score
S2	1:28	0:12	95%
S10	1:55	0:57	89%
S5	2:29	0:56	83% system down, inc
S7	2:30	1:00	69%, inc
S1	3:08	1:01	69%, inc
S3	1:21 System Down	No test	
S6	1:52 System Down, 2nd try	1:00	54%, inc

Averages for CAI subjects:

: 2, 10, 5, 7, 1, 6	=2.2 hr		77%
: 2, 10, 5, 7, 1,	=2.3 hr		81%
: 2, 10	=1.7 hr	.6 hr	92%

Printed CADAM Module One

	Time to complete the Tutorial	Time to complete test	Score
S11	1:28	0:38	91%
S12	1:30	0:42	95% inc missing steps
S8	2:04	0:50	65% inc
S9	2:20	1:00	63% inc
S4	2:55	0:42	89%

Averages for subjects with printed tutorials:

: 11, 12, 8, 9, 4	=2.1 hr		80%
: 11, 12, 4	=2.0 hr	.7 hr	92%

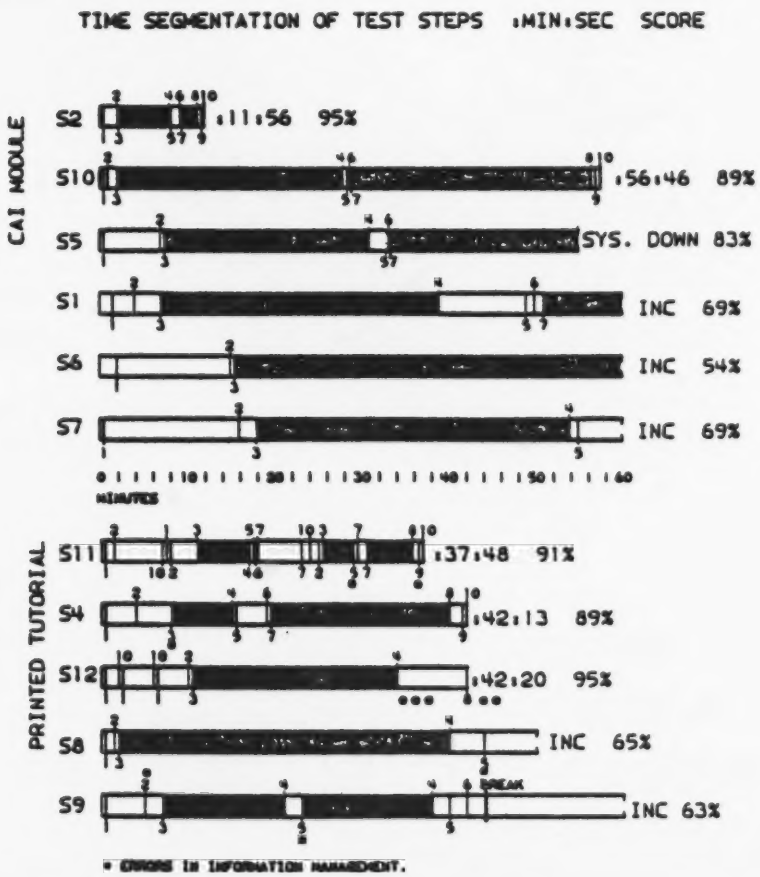
Test scores were plotted against time spent on the tutorial; no pattern or relationship could be detected. The subjects who completed the test and had the highest scores also had the highest and lowest tutorial times. There was no discernible pattern of difference between the performances on test 1 vs. test 2.

Differences

Test scores were also plotted against time spent on the test. The data for the "incompletes" of course grouped near the one hour line. Among those who completed the test there was a slight, inverse relationship between test time and test score. That is, those with the highest test scores completed the test in the shortest times. More subjects would be needed to discern whether this trend is significant. Time segmentation of test steps (See Fig. 24) was achieved by dividing the test into ten non-overlapping steps. (See sample tests in Appendix C). Times for the completion of each of these steps were available from the protocol data. When the sequence of steps was plotted for each subject, and the CAI plots compared with the written, some interesting differences appear.

Figure 24

Time Segmentation of Test Steps



Subjects using the CAI tutorial spent more time than those with the printed tutorial in steps 4 and 8 that involved making and changing drawings, but they had fewer instances of false starts or going back to redo drawings. The longer CAI times seem to be due to longer reference search times for the CAI presentation. In a written tutorial one can flip pages quickly and in bunches; in CAI/CADAM one could search in many ways but screens were always presented one at a time, and there is a refresh cycle of about 2 1/2 seconds.

CAI subjects spent somewhat less time on the information management steps such as windowing and filing and, more important, had far fewer instances of misfiling, misnaming drawings or skipping steps.

Conclusions and Recommendations

From our previous research with documentation for novices we learned that they are very attentive to minute discrepancies. This held true for the CAI documentation as well. It is important that everything be tested, consistent and accurate. Also, new users are often more involved with the CAI system and its operation than with the content of the instruction.

The overall speed and completeness of learning CADAM by CAI seem to be more affected by attitudes and knowledge that the user brings to the learning situation than by differences in presentation. In a group of 12 subjects with similar backgrounds, the individual approaches varied through enthusiasm, curiosity, a sense of duty, and fear.

Tutorials should address the strategy aspects of drawing with CADAM as well as the procedures. For both modes, the major nonproductive delays were caused by gaps in the subject's understanding of the drawing strategy that CADAM requires.

CAI presentation seems to give users a better understanding of the overall sequence of processes required to use CADAM. There were fewer false starts and errors, especially in the information management processes.

If the screen retrieval rate for CAI were faster, there might be an improvement in learning times over written presentation. During early learning there is frequent referencing in either mode. CAI seems to streamline the access to references but the slow retrieval rate apparently cancels the advantage.

Appendix

Appendix A, Interim Report number 3, December 1983

Part II- User Documentation

Introduction

Module One of the CMU CADAM Manual was completed in December 1982 and was well received by IBM. This initial CMU CADAM Manual was found to meet the needs of novice CADAM users--providing clearly defined goals, itemized procedures, and simplified reference aids. The success of the manual was primarily due to the use of Protocol-Aided Revision (PAR) during manual development. (For more information on the PAR methodology and the research used to write and design Module One, see CMU/IBM Usability Study, Interim Report-Number 2, p. 25.)

The initial criteria for Module Two of the CMU CADAM Manual emphasized the needs of the IBM industrial designer as a CADAM user. However when Module One was completed and distributed, it was made available not only to industrial designers but also to IBM engineers. The response from the engineers proved extremely favorable, and they expressed the desire for a more advanced manual from CMU. As a result, the MDA Education Task Force presented CMU with a list of function/concept requirements. The MDA Task Force felt these were the necessary requirements for a training manual used primarily by engineers. MDA asked us to determine if these functions/concepts could be incorporated into Module Two.

We compared the MDA requirements with our criteria for Module Two, function by function. We found that MDA's requirements showed significant overlap with CMU's criteria, approximately 80%. We then proposed to the MDA Education Task Force that Module Two be written and designed with a dual purpose--meeting the needs of both IBM's industrial designers and engineers. We submitted the proposal, and it was accepted by MDA. Our outline for Module Two covering IBM's internal release of CADAM, MDA 20, was revised to include MDA requirements. In particular, these requirements specified a tutorial segment for the Aux View function.

Status

We identify three phases as our workscope for Module Two. In Phase 1, we write and design a complete rough draft, including summaries, references, and index along with graphic layouts and symbols. In Phase 2, we review the draft and conduct a pilot study. Finally in Phase 3, we incorporate protocol analysis into our evaluations and subsequent revisions--alternating between protocol and revision until Module Two is free of user problems. At this time, we are about two-thirds through Phase 1, finishing the complete rough draft.

Schedule

November 1-4	Discuss basic layout, develop outlines for sections and summaries.
November 7-December 2	Develop first drafts of all sections, chapter summaries and reference summaries.
December 5-16	Review and test of manual prototype.
January 23-27	Conduct the first of two protocols.
January 30-February 3	Revise the manual according to protocol results--rewrite, redesign, and mockup new prototype.
February 13-March 5	Cycle between prototype testing and rewriting until the manual has been successfully user tested.
March	Prepare manual for production--final paste-up and copy.

We are now completing the initial rough drafts for each section, writing section summaries, and compiling an index. In addition, we are designing diagrams for each section that explain the software concepts. To insure the graphic quality of the diagrams for Module Two, we have hired a graphic designer. Currently, we are training her on CADAM and familiarizing her with the project in general.

The overall structure of the tutorial has been decided upon. Of course, the structure may change as we run protocols and analyze the results. Our first outline for Module Two defined the manual as a series of drawings. Each drawing centered on a particular function. However, our initial protocols this summer revealed problems with this format. We found that while a drawing provided the user with many opportunities to practice the function, the user's focus was directed towards completing the drawing instead of learning the concepts of the function. The result was that the user found the function concepts difficult to understand and retain. To alleviate this problem, we decreased the amount of

function information that we presented in the drawing and moved the information to a preceding exercise.

This exercise takes the user through the function using a simple sketch. The sketch allows the user to concentrate on the function information because the exercise sketch is less complex than the drawing. In the exercise, the user receives instruction and practice on the function. Whereas in the drawing, he is asked to use what he has learned about the function(s) to create the drawing.

Design

We are improving the graphic format of Module Two by making the current grid more flexible. As we progress through the manual, we are finding that our initial graphic format does not provide us with the flexibility necessary to illustrate more complex functions. These functions have actions sequences of up to fifteen steps as opposed to the less difficult functions whose action sequences average about two to five steps. We are, therefore, testing various graphic formats in conjunction with testing the written drafts.

Equipment

As of November 18, 1983, we are running two CADAM workstations with MDA 20. Due to the lack of software for the 7374 Plotter, we are without hardcopy for the manual drawings. Module Two requires numerous drawings. Without hard copy for the diagrams, structuring page layout and ensuring the accuracy of the manual is extremely difficult and time consuming. The last estimate for having the plotter up and running is approximately mid-December.

Subjects

We will be drawing from IBM employees for subjects. We will represent as much of the CADAM user group at IBM as possible. Protocol testing will begin in late January.

Appendix A. Interim Report number 2, September 1983 Part III - Computer-Aided Instruction

Introduction

This portion of the project is an effort in the area of Computer Aided Instruction, CAI. It involves two activities. The first is to develop parts of a CAI tutorial that parallel specific parts of the written version of our Module Two manual for MDA020 CADAM. The second will be the testing of the package on a variety of subjects with analysis of the data collected during the learning process. Data will help evaluate how well subjects learn using the CAI program, as compared with subjects who used the CMU CADAM manual.

Major System Features of CAI/CADAM

User Friendly: Since the System is going to be positioned to the CADAM workstation, it is imperative that user interaction with the CAI program is kept to a minimum. Therefore, the system is going to require as little keyboard use as possible. Many of the operations will be placed on function keys and these will remain consistent throughout the tutorial.

Menu Driven: The CAI package will be menu driven, allowing the user to move about the tutorial in an unstructured manner.

Help Facility: The program will have a help facility built into it. For most of the screens there will be an additional help screen that the user will be able to access with a particular keystroke. If an additional help screen is not available when the user asks for more information, a general screen of information is displayed.

Glossary: There will be a glossary built into the program. At anytime during the tutorial the user will be able to enter the glossary and ask for a term to be defined.

CAI/CADAM Development Plans

The program is being developed on an XT version of the IBM personal computer with a 10 megabyte hard disk, using MS Pascal and View Manager, a screen development and database management tool. It is currently being developed for a monochrome display with the capability of being run on a color display without change to the application program. We still have one more IBM-XT to purchase. It will have the color graphics board and monitor. We are currently investigating the most appropriate color monitor and board to purchase.

The software system will consist of two major parts, a database of screen images and an

application program. The application program will handle the displaying of screens and the flow of the tutorial. The database will contain a description of each screen needed by the program. It will be accessed by the application program and screens will be displayed according to the user input.

Screens will be developed and stored in a database using View Manager. View Manager is a software package which handles the "painting" of screens and the maintenance of the database in which these screens are stored. A part of the package is a library database. The use of View Manager eliminates the repetitious coding that would otherwise be required by the application program to format each screen and gives us the capability to modify screens without making changes to the application program. Overall, View Manager will reduce the development time of the CAI program.

Accomplishments To Date

The software tools necessary for the development of the CAI program have been purchased and installed on our IBM-XT. These include the MS Pascal compiler and linker, and the View Manager package. Time has been spent becoming familiar with these tools and their capabilities. We are now able to create and store screens using the screen management software, as well as develop and execute Pascal programs.

An initial study of the physical configuration of the many components has been done, as this will certainly be a major usability issue. The IBM-XT will be placed in the same workspace as the CADAM graphics device so that a user can access the CADAM software as the tutorial is moved through. This means that there will be a total of 6 components for the user to handle in sequence. They are the CADAM screen, the CADAM keyboard, the CADAM function keyboard, the CADAM lightpen, the PC screen, and the PC keyboard. The arrangement of these components must be well thought out to insure they do not impair the learning process. We currently feel that the PC screen should be placed above the CADAM screen and angled toward the user. It is not yet clear what position the PC keyboard should take, although it is detachable and the user may just want to place it in his lap. More investigation will be done in order to determine the final configuration of all the components.

Time has been spent consulting with a man-machine interface expert, Dr. K. Robertson, to determine the most effective page layouts and techniques for such tutorials. Dr. Robertson has conducted research on the interaction of novice and expert users with CAI programs. The results show, among other things, that in order for a CAI system to be beneficial the user should be presented with manageable chunks of information, the most important information should be located at the top of the screen, the number of lines of text on a screen should be eight or less, and contextual information should be provided on each screen to let the user know where he or she is in the tutorial. Through this research the

path of a user's eye movement has also been determined. During those times when a user becomes confused and does not know what to do next expert users scan the perimeter of the screen in a clockwise direction, while novice users tend to stare right at the middle of the screen. Currently, we are trying to incorporate all of this information into the CAI system and the creation of its screens.

A prototype of the CAI program is being developed to illustrate the way in which we intend the program to work. It will demonstrate the menu structure of the program and contain a lesson on one element of the CADAM software.

Introduction

As we said at the beginning of this report, good documentation satisfies a short term need for improving the usability of current systems. As systems become more friendly and less dependent on user aids, there will remain a need for good documentation, either online or hardcopy. By developing design principles through research, we can provide guidelines for future documents.

Our goal in this section of the report is to describe the research conducted at CMU to develop Modules One and Two of our CADAM Manual.

Background

In the fall of 1981, the first draft of Module One was written at CMU with the help of existing documentation but without equipment and software. CADAM was installed at CMU in 1982, and in December the prototype version of Module One was released. This manual is being successfully used for self-study by faculty, staff, and students. To date 3 faculty, 45 students as well as others have used the manual to learn the basics of CADAM. Since June 1983 the Department of Design and the Communications Design Center have been developing Module Two for MDA 20 software release.

Part III - User Documentation

Manual Development - Design Considerations

In planning the overall structure of our manual we considered the experience level of the user and the type of documentation that would best suit the users' needs. This strategy represents an important departure from manual development elsewhere because it focuses on the audience for the manual. Based on level of experience, users can be categorized as novice, occasional, frequent. Each category requires a different type of documentation (Figure 7).

Figure 7, User Documentation Needs

Document Type	User Group		
	Frequent	Occasional	Novice
Tutorial		○	●
Reference Cards		●	●
Reference Guide	○	●	●
Technical	●	○	

●=needed frequently

●=needed occasionally

Our user group consists of industrial designers and mechanical engineers. Within IBM the number of CADAM users has increased dramatically in the last year and is continuing to do so. But the number of CADAM workstations available to the number of users means that most users will fall into the Occasional category. Although our manual is aimed primarily at the needs of the novice, a novice user is only a novice for a short period of time. Thus, our manual also meets the needs of the occasional user, by including reference information as well as lessons.

Manual Development - Design Principles

We have identified, through research, research literature, observations, and experience in creating manuals, six basic principles for manual development. Implicit in our research methodology is the use of PAR for testing and revising manuals. The findings from PAR were essential for identifying the basic principles and applying them as guidelines to create our first drafts.

Develop Manual Around Needs of Particular User Group--Often programmers write manuals for novices. However, because the programmer does not understand the novice's viewpoint, such manuals are often confusing and frustrating to the novice. Thus, the different needs of a particular user group must be incorporated into the manual along with the software applications. In this way, the user can learn from the manual and then apply his learning.

Develop Concise and Consistent Language--Special terms should be defined and used consistently. The text should be as concise as possible without sacrificing clarity. The style should be straightforward and positive but not too informal or "cute." Most of all language should be clear and instructions accurate.

Develop Applied Lessons as Examples--The lessons should be similar to problems that the user will encounter on the job.

Develop Easily Understood Reference Information--All users must be able to locate information in the document easily. A Table of Contents and an Index are useful. However, we have found that additional clues must be incorporated into the design of the manual to help the user locate information. We have developed page headings and other graphic features to highlight certain types of information.

Develop Consistent Graphic Layout--The general format of the manual should be consistent throughout. This helps the user understand how to use the document. Typographic information should have a clear hierarchy. Diagrams and symbols should help to improve communication of the material being presented.

A Brief Review of Module One 1981-1982

Our manual for the CADAM system consists of two modules. Module One focuses on the CADAM hardware and the basic construction functions (POINT, LINE, CIRCLE, CORNER, SPLINE). Module Two focuses on the CADAM software and the more complex functions (WINDOW, RELIMIT, OFFSET, DIMENSION, GROUP, DETAIL, AUX VIEW).

Module One was developed using Protocol-Aided Revision (PAR). PAR begins with an initial draft and consists of three basic steps: video tape and record subjects using initial draft; analyze protocol for evidence of user difficulties; revise draft based on analysis of protocol; then recycle. This process is repeated through several cycles until subjects demonstrate no difficulties in using the manual. The procedure is carried out by writers who have been trained to identify the reasons for the user's difficulties. PAR has proven to be an effective editing tool, helping writers produce well-designed, clearly written, usable documents.

By analyzing the protocols collected during Module One development, we clarified our assumptions about what kinds of information novices need. Briefly, novice users need clearly defined goals, both global and local. They must be given a context in which to work. Instructions must be presented as itemized procedures and new or unfamiliar terms must be defined when first presented. Novices must be reinforced and encouraged to experiment while they are learning. However, they must be provided with immediate information that will help them recover from errors. Finally, novices need aids for review and reference. (For more information, see Final Report of December 15, 1982.)

The prototype manual, Module One, for the CADAM system meets these needs. Module One contains a **Table of Contents** to give the user a concise overview of what the manual included. A short **Introduction** explains how to use the manual and describes the CADAM equipment. Module One includes **Exercises** that provided procedural instructions along with **Summaries** at the end of each Exercise that provide a quick review and later reference. An **Abbreviations Guide** allows the user to understand the information in the message area. Finally, a **HELP Section** provides information for error recovery.

Module Two Development 1983

To date we have used PAR to test the initial draft of Module Two on four subjects. Module Two development is based on the results of our protocol analysis, the principles developed from Module One, our knowledge of CADAM and our own expert judgments.

Like Module One, Module Two has a clearly defined structure. The **Table of Contents** is based on the information hierarchy of the manual. The **Introduction** is as brief as possible and provides continuity by explaining changes in the graphic and written format. It also explains basic concepts of the software (such as Window area, Drawing Space, etc.). The **Drawings and Exercises** allow the user to learn the new functions and review as necessary. The **Summaries** at the end of each Drawing or Exercise allow the user to review the material just presented.

In addition to the above, our plan for Module Two includes four added features:

> **The Expert Strategies Section**--dedicated to all the things the system never told you. It is a collection of tips, shortcuts, and tricks that users have told us about or we have discovered.
>
> **The Reference Summaries**--designed to help the initiated novice and the occasional user. It can be used while the novice is training or afterwards, if he forgets a procedure. Reference Summaries are organized in a how-to format and include procedures that may not have been covered in the drawings or exercises. This section can either remain as part of the manual or be separated by the user.
>
> **The Index**--includes commands and menu selections as well as "how to" statements. This gives the novice, occasional, and frequent users equal access to needed information.
>
> **The Pocket Reference Cards**--will contain information on; logon and logoff procedures, plotting procedures, function summaries and abbreviations.

The exercises in Module One were fundamental and based on the construction of elements within a function, which were appropriate for introducing the novice user to CADAM. However, the functions covered in Module Two are more complex, it was clear to us that simple exercises alone could not teach these functions. We address this difference by giving the user more challenging and interesting assignments, which we call Drawings.

The Drawings in Module Two are of real objects and, in general, represent objects similar to those the user will be drawing on CADAM after training. They are designed to help the user learn the interdependency of the functions. However, each Drawing focuses on a particular function.

After testing the initial drafts of Module Two, we found that while the users could work through the Drawing successfully, the information was too complex to facilitate learning. To solve this problem, we added a series of Exercises to Module Two. These Exercises employ simple abstract shapes, allowing the user to concentrate on learning the procedure instead of completing the Drawing. In addition, the Exercises provide another level of review. That is, a user may work through a Drawing, work through an Exercise, or refer to a Summary--depending on how much information is needed.

Within each Drawing we established a hierarchy of information.

> Start-up--describes the objective of the Drawing, the steps to complete it, and the parameters for starting it (such as units, drawing name, etc.)
>
> Steps--provide an itemized list by which a Drawing may be completed.
>
> Plan--proceeds each Step and explains how the Step will be carried out.
>
> Action Brief--breaks up information into manageable chunks. Because each Step in a Drawing can consist of up to 20 Action/Response units, the user can become lost in the process. Thus, the Action Brief helps to reorient the user as to why the Actions are being performed. Action Briefs are not labeled as such but are highlighted graphically.
>
> Action/Response Units--tells the user what Action to perform and what the Response will be. Action/Response Units are grouped graphically and are clear, concise, accurate.
>
> Function and Menu Selection Overviews--describe the purpose and use of the function or menu item. Menu selection overviews are similar to function overviews, but they focus on a specific procedure within a function.

Graphic Design

The graphic format of Module Two is modeled after Module One. However, we have made some changes in order to clarify symbols, improve usability, and accommodate diagrams required by the more complex functions. As we continue to develop Module Two, our process of refinement may require additional changes to accommodate the subject matter and improve usability.

Our primary goal in making the following design changes was to reduce the amount of text the user must read and thus facilitate working through the manual. We abbreviated the labels Action and Response to A and R. A and R appear in boldface type and are hung from the first line of text. This format flags each unit and reduces reading and search time.

In addition to the A and R changes, we experimented with line widths. We consulted with a graphic designer, reviewed existing research, and carefully modified line length and leading (the spacing between lines of text) to make the text easier to read and to reinforce the levels of information.

We also incorporated the same visual symbols as in Module One, but we refined the symbols for Select and Indicate. In addition, as the user becomes more proficient, we are experimenting with dropping extraneous symbols or words. We are using the protocols to determine at what point to make such changes.

Part III - User Documentation

Summary

One major difference between our CADAM Manual and others is that it functions as both a tutorial and a reference manual. That is, once a novice works through the drawings and exercises, other features of the manual help the user to locate information quickly for review and reference. This is important because the majority of the users on the CADAM system are in the occasional category.

The most significant feature of our document design process is that we use PAR to evaluate and revise the manuals. This technique ensures that the manual will meet the needs of the user. The manual is visually attractive which has a positive effect on the user's attitude as well as increasing the usability of the manual.

Usually a trainee is given one work week (40 hours) to learn the basics of the CADAM system. The trainees work with instructors who provide classroom lectures as well as individual attention. Our Goal is to keep the time required to go through both manuals between 30 to 40 hours. Module One takes from 25 minutes to 2 hours to complete, depending on the user's experience level. Users who have previous experience with computer and or graphics systems may require less time. Modules One and Two are designed to be used for self-study, thus reducing the amount of instructor time required. This is especially useful in remote sites where instructors are not available.

Appendix B

The CADAM CAI Tutorial Prototype Program
Technical Documentation

A Prototype Program
Written for the IBM Installation
at Carnegie -Mellon University

Produced by the
Department of Design
Carnegie -Mellon University
Pittsburgh, PA 15213

Author: Lori Oncea Haley
Date: April 1984

Table of Contents

1. Introduction
2. Program Implementation Information
3. Program Overview
4. How to Compile and Link the Program
5. How to Edit Screens and Organize the Screen Databases
6. The Files and Their Contents
7. The Subroutines
8. The Screen Database Files
9. The Database Screen Names
10. The Data Structures
11. The Data Structure Diagrams
12. How the Data Structures are Initialized
13. The Program's Main Algorithm
14. Helpful Things to Know
15. Conclusion

Introduction

The CADAM CAI Tutorial is a computer aided instruction program which teaches the use of MDA020, IBM's internal version of CADAM. The tutorial was designed to run on a PC-XT located next to a CADAM workstation, requiring only the use of the 10 function keys located on the left side of the PC-XT keyboard for operation. The program is menu driven and has various reference aids built into it. All the instructions for using the tutorial are explained in the beginning of the CAI program.

This is a prototype program which was developed in order to compare computer-aided instruction learning of CADAM with learning from printed manuals. It is, therefore, a test program and is not meant to be expanded upon if a full scale CADAM tutorial was to be developed. A comprehensive tutorial would be best if it ran on the same hardware as CADAM and was more closely integrated with the CADAM software.

The technical documentation that follows is provided for someone who might want to modify or enhance the prototype. It describes the technical aspects of the CADAM CAI system. The easiest way to understand how the program works is to run the program, skim through the documentation and then look at the code along with the documentation.

Program Implementation Information

The tutorial was developed on an IBM PC-XT with 256K memory and a monochrome display. It is written in MS-PASCAL using two development productivity packages. PASCAL Tools version 2.4 and View Manager version 1.6. Both packages are products of Blaise Computing Inc., 1609 Acton Street, Berkeley, California. MS-PASCAL is a product of Microsoft Corporation.

View Manager is a software package which handles the "painting" of screens and the maintenance of the databases in which the screens are stored. A part of this software is a library of routines that are used to access the databases from the tutorial program. The use of View Manager eliminates the repetitious coding that would otherwise be required by the tutorial program and gives the capability to modify screens without changing the program.

Program Overview

The program is most easily understood by running through the tutorial. Before reading further, we recommend running through the tutorial to become familiar with the user interface.

The program consists of two major parts, several databases of screen images and an application program. The application program handles the presentation of the screen images, reads user input, and controls the flow of the tutorial. Each screen needed by the program is located in one of the screen databases. The presentation order fo the screens depends on user input and the way the data structure has been initialized.

There are four types of screens used in the tutorial: instruction screens, menu screens, scroll screens and text screens. Each type of screen includes prompts for the use of the function keys. All program operations are performed according to which function keys the user presses. For each screen only certain function keys are active for input, if the user presses an inactive key, the system will sound a short beep, indicating that it is not an active function key. On instruction and reference screens the valid function keys are shown at the bottom of the screen. For menu and scroll screens these keys are shown in the central area of the screen. (See ----- for screen layout info)

There are three global functions, GLOSSARY, INDEX, and HELP, associated with the keys F7, F8, F9 respectively. They are in effect any time you are not in that particular function, there is no recursion. For example, if the user presses F7 and enters the Glossary he may not press F7 again and get into glossary again until he exits from the Glossary.

How to Compile and Link the Program

The program is compiled and linked to form the executable module by running the batch job CAIC.BAT. The job calls a second batch file, CAIL.BAT. A look at these ASCII files will show you how the MS-PASCAL compiler and linker are called.

How to Edit Screens and Organize the Screen Databases

Both of these tasks are done with the View Manager software. The View Manager manual has complete documentation for using the screen editor and database reorganizer programs. Each database used with the tutorial contains the five screens necessary to use the screen editing program. They are SSUTIL01 - SSUTIL05 and you will see them in the list of database screen names for each database.

The Files and Their Contents

CAI.PAS	The main program for the tutorial. It also contains procedures and functions for the help, index and glossary features of the program, as well as ones for user input and database checking. The routines are: CAI-CADAM (main), CLRDB, SCRIDX, SCRGLS, READSN, HELP, and INDEX.
CAIGLB.INT CAIGLB.IMP	Interface and implementation files for the global variables. See code for variable names and documentation.
CAISCN.INT CAISCN.IMP	Interface and implementation files for auxiliary routines. Most of these routines are used to initialize the data structures. One is used to retrieve the next screen which should be presented. The routines are: HASH, INSERT, ASSOC, ASSODB, GETNEXT, IDXADD, and GLSADD.
CAIINT1.INT CAIINT1.IMP	Interface and implementation files for the initialization routine. This routine makes all the routine calls which set up the screen data structure. The routines are: INTSCN.
CAIINT2.INT CAIINT2.IMP	Interface and implementation files for the other initialization routines. These routines make the calls to initialize the data structures for the index, glossary, and database location for each screen. The routines are: INTDB, INTIDX, and INTGLS.
DATABASE FILES	Database files containing the screens for the tutorial. Please see the section about data-bases for a complete description of which screens are located in each database. CAICRNIX.BCI CAICRNFF.BCI CAIOTHIX.BCI CAIOTHFF.BCI CAIPNTIX.BCI CAIPNTFF.BCI
CAIC.BAT	Batch file used to compile and link the CAI program together.
CAIL.BAT	Batch file called by CAIC.BAT. This file has the command for the linker in it.
NAMES.EXE	This is program used to get out a list of screen names from a database without using the View Manager software.

The Screen Database Files

The following databases consist of two DOS direct access files, index file and a field information file. The index file contains screen names and pointers to the field information file where the field information for each screen is found. If the screen database name is SDBNAM, the file names are:

SDBNAMIX.BCI	Index file
SDBNAMFF.BCI	Field information file

DATABASE NAME	CONTENT DESCRIPTION
1. CAIOTH	contains menu screens including the HELP menu, the titlepage screen, the INDEX screen, the GLOSSARY screen, the screens for How To Use The Tutorial, the screens for the CADAM Overview section, and the HELP screens.
2. CAIPNT	contains most of the screens for Exercise 1, how to construct points in CADAM.
3. CAILIN	contains all the screens for Exercise 2, how to construct lines in CADAM.
4. CAICRN	contains all the screens for Exercise 3, how to make a corners with CADAM, and the remaining screens for Exercise 1, how to construct points.
5. CAIWND	contains all the screens for Exercise 4, how to use the window function, and all the CADAM summaries screens, including the summary menus.

The names of the screens contained in each of these databases can be found in the next section of this document.

The Database Screen Names

The screens contained in each database are listed here by database. If the first five files listed for a database are SSUTIL01 – SSUTIL05, you may copy the database files to SCREEN*.BCI and use VIEWSCN to access the screens. If not, this is an application database only and can NOT be used with the VIEWSCN program.

CAICRN:

The files for this screen database are CAICRNIX.BCI and CAICRNFF.BCI. There are 53 screens in the database CAICRN. They are:

1. SSUTIL01
2. SSUTIL02
3. SSUTIL03
4. SSUTIL04
5. SSUTIL05
6. CORN03
7. CORN01
8. CORN02
9. CORN04
10. CORN05
11. CORN06
12. CORN07
13. CORN08
14. CORN09
15. CORN10
16. CORN11
17. CORN12
18. CORN13
19. CORN14
20. CORN15
21. CORN16
22. PNT057
23. PNT058
24. PNT059
25. PNT060
26. PNT061
27. PNT062
28. PNT063
29. PNT064
30. PNT065
31. PNT066
32. PNT067
33. PNT079
34. PNT068
35. PNT069
36. PNT070
37. PNT071
38. PNT072
39. PNT073
40. PNT074
41. PNT075
42. PNT076
43. PNT077
44. PNT078
45. PNT080
46. PNT081
47. PNT082
48. PNT083

CAILIN:

The files for this screen database are CAILINIX.BCI and CAILINFF.BCI.
There are 55 screens in the database CAILIN. They are:

1. SSUTIL01
2. SSUTIL02
3. SSUTIL03
4. SSUTIL04
5. SSUTIL05
6. LINE01
7. LINE02
8. LINE03
9. LINE04
10. LINE05
11. LINE06
12. LINE07
13. LINE2A
14. LINE2B
15. LINE2C
16. LINE7A
17. LINE7B
18. LINE08
19. LINE09
20. LINE10
21. LINE11
22. LINE12
23. LINE13
24. LINE14
25. LINE15
26. LINE16
27. LINE17
28. LINE18
29. LINE19
30. LINE20
31. LINE21
32. LINE22
33. LINE23
34. LINE24
35. LINE25
36. LINE26
37. LINE27
38. LINE28
39. LINE29
40. LINE30
41. LINE31
42. LINE32
43. LINE33
44. LINE34
45. LINE35
46. LINE36
47. LINE37
48. LINE38
49. LINE39
50. LINE40
51. LINE41
52. LINE42
53. LINE43
54. LINE44
55. LINE45

CAIOTH:

The files for this screen database are CAIOTHIX.BCI and CAIOTHFF.BCI.
There are 48 screens in the database CAIOTH. They are:

1. SSUTIL01
2. SSUTIL02
3. SSUTIL03
4. SSUTIL04
5. SSUTIL05
6. HELP01
7. HELP02
8. HELP03
9. HELP20
10. HELP21
11. HELP22
12. HELP23
13. HELP24
14. MAINMN
15. CADMEN
16. TITLEP
17. HELPMN
18. INDEX1
19. GLOSS1
20. HOW001
21. HOW002
22. HOW02A
23. HOW003
24. HOW004
25. HOW005
26. HOW006
27. HOW007
28. HOW07A
29. HOW009
30. HOW012
31. HLPIDX
32. OVER01
33. OVER02
34. OVER03
35. OVER04
36. OVER05
37. OVER07
38. OVER08
39. OVER10
40. OVER09
41. OVER11
42. OVER12
43. OVER13
44. OVER14
45. HOW010
46. HOW011
47. OVER15
48. HOW01A

CAIPNT:

The files for this screen database are CAIPNTIX.BCI and CAIPNTFF.BCI.
There are 62 screens in the database CAIPNT.
They are:

1. SSUTIL01
2. SSUTIL02
3. SSUTIL03
4. SSUTIL04
5. SSUTIL05
6. PNT001
7. PNT002
8. PNT003
9. PNT004
10. PNT005
11. PNT006
12. PNT007
13. PNT008
14. PNT009
15. PNT010
16. PNT011
17. PNT012
18. PNT013
19. PNT015
20. PNT12A
21. PNT016
22. PNT014
23. PNT017
24. PNT018
25. PNT019
26. PNT020
27. PNT021
28. PNT022
29. PNT023
30. PNT024
31. PNT025
32. PNT026
33. PNT027
34. PNT028
35. PNT029
36. PNT030
37. PNT031
38. PNT032
39. PNT033
40. PNT034
41. PNT035
42. PNT036
43. PNT037
44. PNT038
45. PNT039
46. PNT040
47. PNT041
48. PNT042
49. PNT043
50. PNT044
51. PNT045
52. PNT046
53. PNT047
54. PNT048
55. PNT049
56. PNT050
57. PNT051
58. PNT052
59. PNT053
60. PNT054
61. PNT055
62. PNT056

CAIWND:

The files for this screen database are CAIWNDIX.BCI and CAIWNDFF.BCI.
There are 61 screens in the database CAIWND. They are:

1. SSUTIL01
2. SSUTIL02
3. SSUTIL03
4. SSUTIL04
5. SSUTIL05
6. WIND01
7. WIND02
8. WIND03
9. WIND04
10. WIND05
11. WIND06
12. WIND07
13. WIND08
14. WIND09
15. WIND4A
16. WIND48
17. WIND10
18. WIND12
19. WIND11
20. WIND13
21. WIND14
22. WIND15
23. WIND16
24. WIND17
25. WIND18
26. WIND19
27. WIND20
28. WIND21
29. WIND22
30. WIND23
31. WIND24
32. WIND25
33. WIND26
34. WIND27
35. WIND28
36. SUMMN1
37. SUMMN2
38. SUM001
39. SUM002
40. SUM02A
41. SUM02B
42. SUM02C
43. SUM02D
44. SUM003
45. SUM004
46. SUM005
47. SUM006
48. SUM007
49. SUM008
50. SUM009
51. SUM010
52. SUM012
53. SUM013
54. SUM014
55. SUM015
56. SUM016
57. SUM017
58. SUM011
59. HLPSM1
60. HLPSM2
61. WIND29

The CAI Subroutines

This is a list of all the routines and the ascii files where the routines can be found. Please see the code for complete documentation of what each procedure or function does.

Name	File Location	Brief Purpose Definition
ASSOC	CAISCN.IMP	Builds linked list of screen - function key associations.
ASSODB	CAISCN.IMP	Associates a database name a screen.
CAI-CADAM	CALPAS	Main program.
CLRDB	CALPAS	Opens the needed database if it is not already open.
GETNEXT	CAISCN.IMP	Gets next screen to display.
GLSADD	CAISCN.IMP	Initializes glossary list.
HASH	CAISCN.IMP	Does hash for symbol table location based on screen name.
HELP	CALPAS	Handles the HELP feature.
IDXADD	CAISCN.IMP	Initializes the index list.
INDEX	CALPAS	Handles the INDEX feature.
INSERT	CAISCN.IMP	Puts a screen name in to the symbol table.
INTDB	CAIINT2.IMP	Makes calls to initialize the database pointers.
INTGLS	CAIINT2.IMP	Makes calls to initialize the glossary list.
INTIDX	CAIINT2.IMP	Makes calls to initialize the index list.
INTSCN	CAIINT1.IMP	Makes calls to initialize the symbol table with screen names.
READSN	CALPAS	Reads user input.
SCRIDX	CALPAS	Handles scrolling for index feature.
SCRGLS	CALPAS	Handles scrolling for glossary feature.

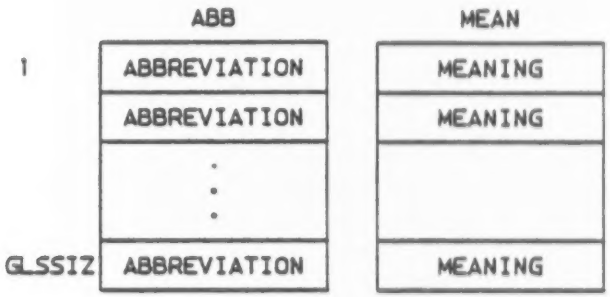
The Data Structures

The main data structures are the symbol table, the database list, the index word list, and the glossary abbreviation and meanings list. The variables for these structures are defined in the global variable interface file, CAIGLB.INT. The names can also be found in the data structure diagram. This diagram along with the following general description of each data structure explains the fundamental structures of the tutorial program.

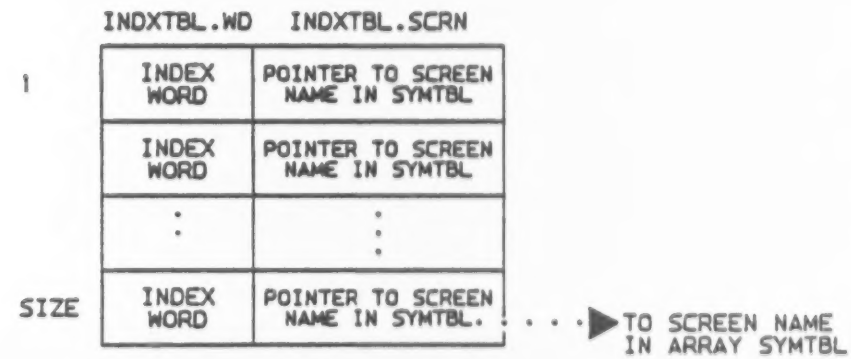
Symbol Table	The symbol table holds the name of each screen used in the tutorial. For each entry in the table there is also a pointer to a linked list and a key for the database the screen is kept in. The linked list is a list of elements indicating the valid function keys for this screen and which is the next screen to present if this key is selected by the user. The global function keys, F7, F8 and F9, are not put into this linked list since they are the same for all screens. The key for the database the screen is kept in is an index into the database name array.
Database List	This is an array constant which holds the names of databases used by the tutorial.
Index Word List	This is a list of the words that comprise the index. For each word in the index there is a key to the screen where the word is defined. This key is an index into the symbol table array. It points to the location in the symbol table of the screen the word is found on.
Glossary List	The list for the glossary is actually two arrays, one for the abbreviation and one for the meaning of the abbreviation. For each entry in the abbreviation array there is a corresponding entry in the meaning array.

The Data Structure Diagrams

The Glossary Abbreviation and Meaning Lists

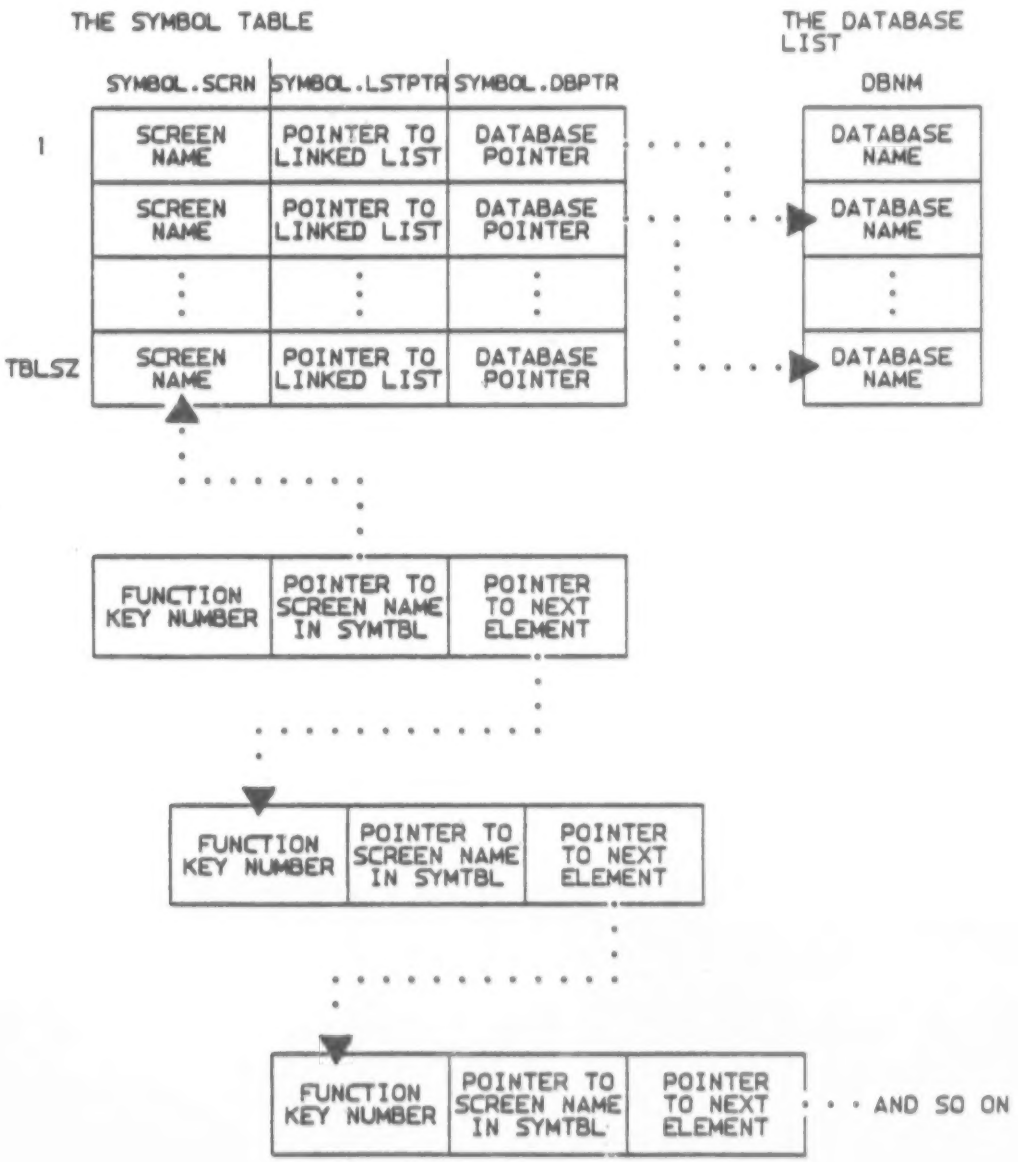


The Index Word List



The Data Structure Diagrams

The Symbol Table and Database List



How The Data Structures Are Initialized

In order for the program to run it has to have a certain amount of knowledge. This knowledge is kept in its data structures. The program gets knowledge when these structures are initialized. The following section will describe what needs to be initialized and where the initialization is done. You will need to understand this if you wish to change the sequence of screens, add or delete screens, change the index or glossary list, change a screen's database location, or add or delete databases.

The tutorial program has certain subroutines which do the data structure initializations. They are ASSOC, ASSODB, INSERT, GLSADD, and IDXADD. The calls for these subroutines are all located in a few routines which the main program calls. These routines are INTDB, INTGLS, INTIDX, and INTSCN and they are the ones you must alter in order to change the programs knowledge.

INTDB	This routine makes all the calls which initialize the symbol table database location for each screen. For every screen in the tutorial, INTDB makes a call to ASSODB passing it a screen name and the database the screen is stored in. INTDB uses integer constants which index into the database array where the database names are kept.
INTGLS	This routine makes all the calls to initialize the glossary abbreviation and meaning lists. For every abbreviation in the glossary it calls GLSADD, passing it the abbreviation and its meaning. If more words are added to the glossary, you may have to change the size of the glossary array by increasing the global variable GLSSIZ.
INTIDX	This routine makes all the calls to initialize the index word list. For every word in the index it calls IDXADD, passing it the index word and the name of the screen where the index word is defined. If many new index words are added, the size of the global index word array may have to be increased by changing the global variable SIZE.
INTSCN	This routine initializes the symbol table screen names and the pointer to the linked list. For each function key, other than the global ones (F7, F8, F9), which is a valid input for this screen it makes a call to ASSOC, passing it the screen name, the function key, and the name of the screen the program should present if the user chooses this function key. This is the key to how the program knows where to travel next. For example, F2 always takes you forward a screen, so if screen EXAMP5 should follow screen EXAMP4, the call would be: ASSOC('examp5',f2,'examp4');

The Program's Main Algorithm

Generally, the tutorial program works as follows. The program displays a screen to the user, waits for user input and then displays the next screen according to the input given. You will see in CADAM_CAI (main) that a CASE statement imbedded in a DO WHILE statement is used to control program flow. Each of the global functions, GLOSSARY, INDEX and HELP, has its own index in the CASE statement and is implemented in a subroutine located in the same file as the main program. For all other function keys the system calls GETNEXT to see which screen to display next. GETNEXT checks the current screen's linked list to see which screen it should display next. If the answer is not in the list, the program assumes it received an invalid input, beeps, and waits for new input. This loop continues until the flag is set indicating to exit the program.

Helpful Things to Know

The global functions GLOSSARY, INDEX, and HELP each have their own subroutines. Unfortunately, because the program was modified after the original design was laid out, some of the generality was lost and things were hardcoded into the HELP routine to make it operate properly. Do not fear though, the program is just not that big! I will try to layout all the special case areas here.

Two of the functions on the HELP menu, F4 and F3, use the same screens that are used in the normal flow of the tutorial. Since the linked list for these screens contains information to be used only during the normal flow, the HELP routine had to be coded to know when not to use the information from the linked list. Specifically, for the first screen and the last screen of a sequence the F1 and F2 next screen, respectively, from the linked list would indicate the wrong screen to go next when the user was in the HELP function. So, the HELP routine knows the names of these screens and acts differently when it sees them. Also, since F3, 'Help with CADAM Procedures', takes you to a set of screens which has menus, new menus were added to be used when in the help function. The names of these menus and the names of the normal sequence menus are hardcoded into the HELP routine. Because of these special cases, if the CADAM Summaries menus are changed or the topics are expanded, the HELP routine would have to be changed to accommodate this.

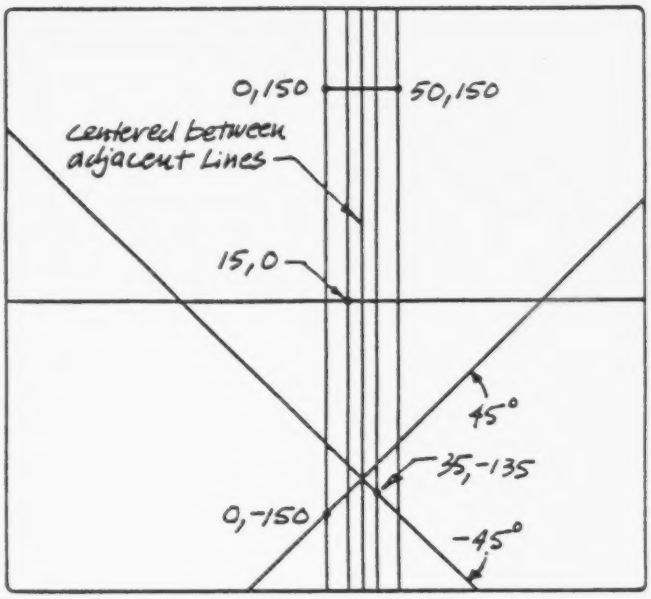
Conclusion

There is still some room for the tutorial to be expanded but not as much as would be needed a complete CADAM tutorial to run on the PC-XT. If the size of the various arrays are increased you may run into problems with the compiler running out of room. We a little modification you might be able to remedy these problems. Then there is the linker to deal with. It currently creates a VM.TMP file for additional memory to get the program linked. Given the size of CADAM, a comprehensive tutorial would be much too big for a PC-XT.

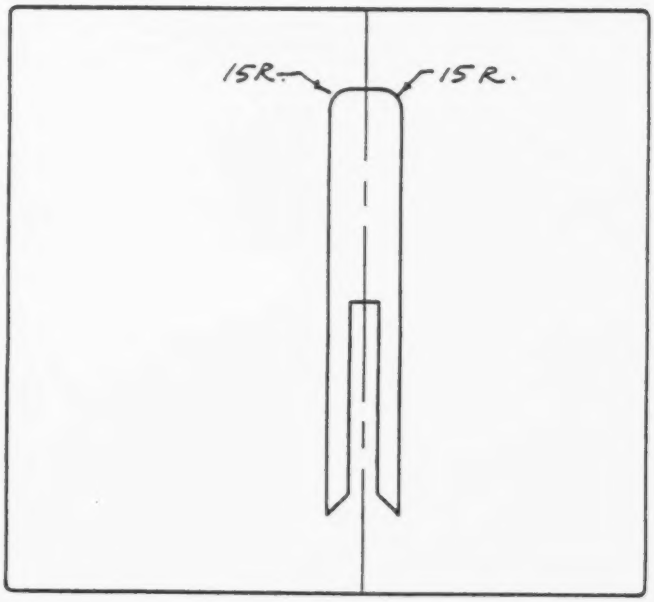
Appendix C

TEST 1

1. Logon to CADAM.
2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials.
3. Go to Window Size = 1.5
4. Create the drawing shown to the right.
5. When it looks like this, FILE the drawing.

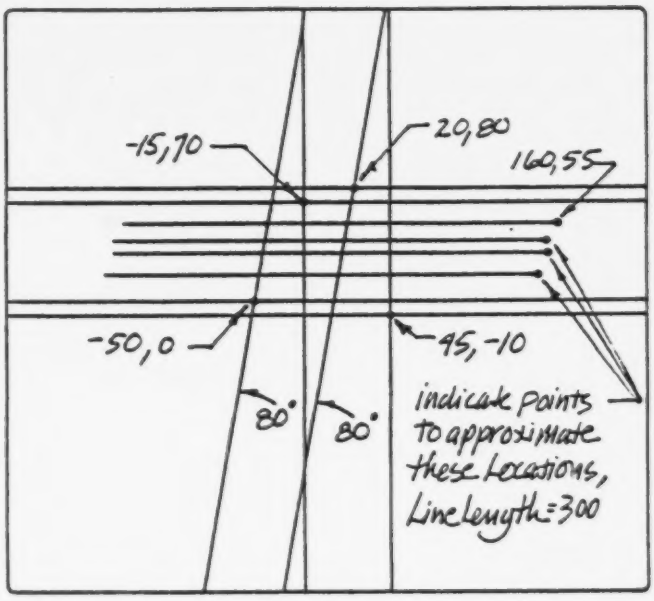


6. Call up the list of drawings.
7. Call back the drawing you just did, C.P. GRID,XXX.
8. Change the drawing as shown to the right.
9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX.
10. Logoff your User ID only.

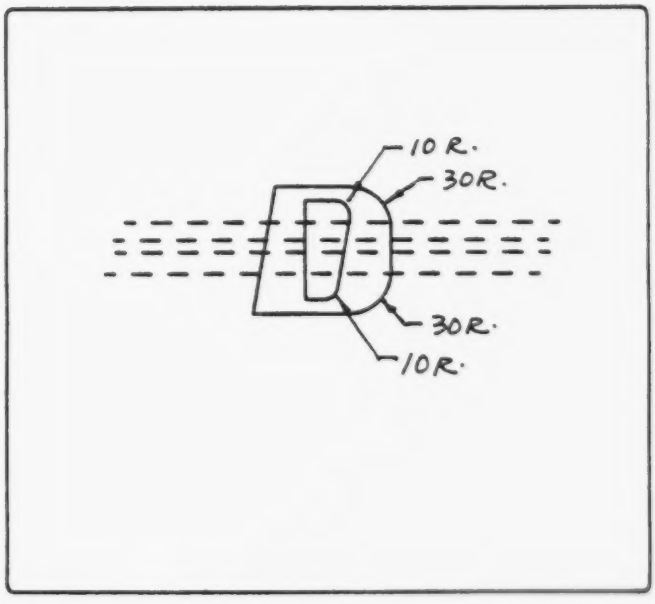


TEST 2

1. Logon to CADAM.
2. Start a drawing with the Part Number D.L. GRID and the Find Number XXX (your initials.
3. Go to Window Size = 1.5
4. Create the drawing shown to the right.
5. When it looks like this, FILE the drawing.



6. Call up the list of drawings.
7. Call back the drawing you just did, D.L. GRID,XXX.
8. Change the drawing as shown to the right.
9. When it looks like this, rename and file the drawing as: DASH LOGO,XXX.
10. Logoff your User ID only.



Subject 1 Mpl. 1. #1

Scoring Sheet

TEST 1

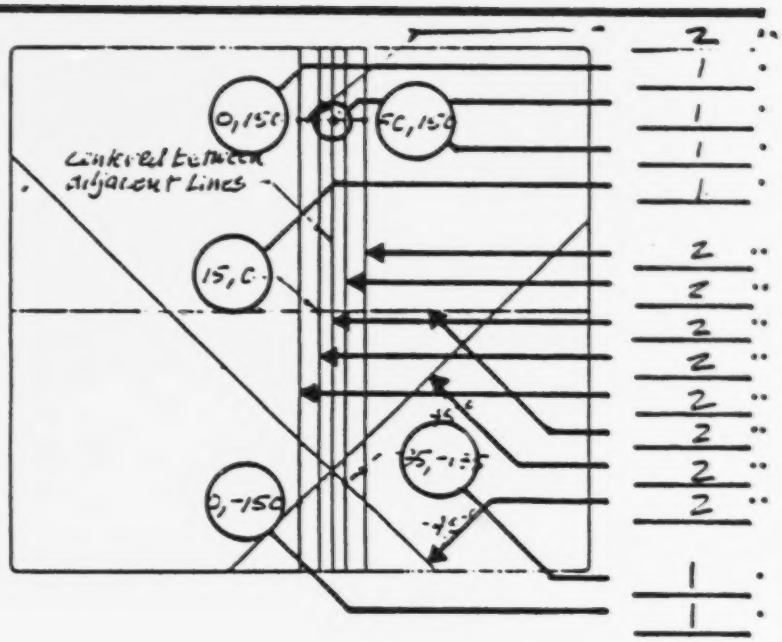
:03:10 1. Logon to CADAM

:05:54 2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials.

:08:37 3. Go to Window Size = 15

:41:17 4. Create the drawing shown to the right

:51:35 5. When it looks like this, FILE the drawing



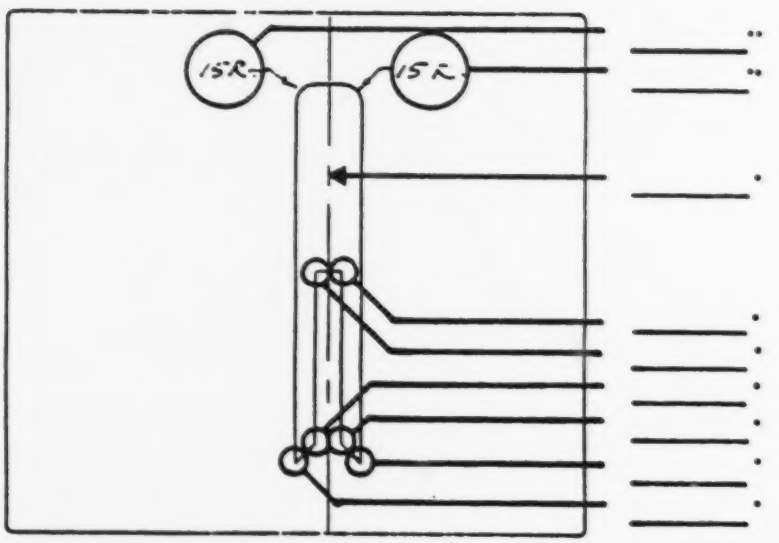
:52:56 6. Call up the list of drawings.

:54:31 7. Call back the drawing you just did. C.P. GRID,XXX.

Stopped. 62:56 8. Change the drawing as shown to the right.

_____ 9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX.

_____ 10. Logoff your User ID only.



:01:41 B test

1:01:14 T test

3:08: T tutorial (C) W

Score 24 /35

% 69

61:51
1:
=61:14

Subject 2 Tajeu 1 #2

Scoring Sheet

TEST 2

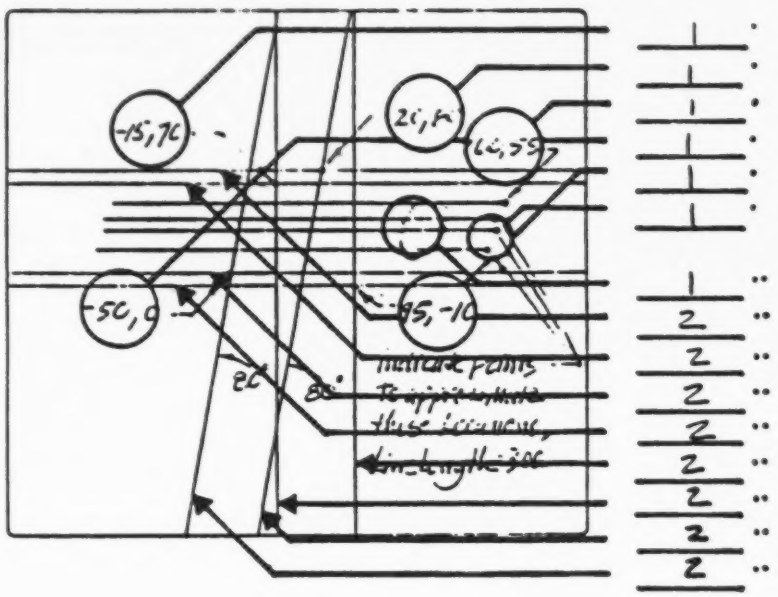
:0:23 1. Logon to CADAM

:2:10 2 Start a drawing with the Part Number D.L. GRID and the Find Number XXX (your initials.

:2:22 3 Go to Window Size = 15

:8:27 4 Create the drawing shown to the right.

:8:43 5 When it looks like this, FILE the drawing.



1 1 1 1 1 1
1 2 2 2 2 2 2 2 2

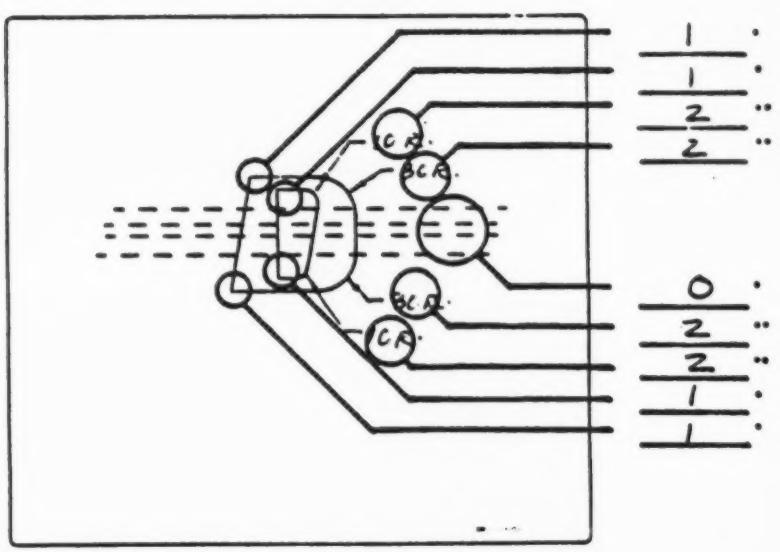
:9:02 6. Call up the list of drawings

:9:07 7. Call back the drawing you just did, D.L. GRID,XXX.

:11:25 8. Change the drawing as shown to the right.

:11:54 9. When it looks like this, rename and file the drawing as: DASH LOGO,XXX.

:12:04 10. Logoff your User ID only.



1 1 2 2
0 2 2 1 1

:0:08 B test

:11:56 T test

1:28: T tutorial (C) W

Score 35 /37

% 95

102 35

Subject 3

Scoring Sheet CADAM Hung up @ 1:21, NS Test

TEST 1

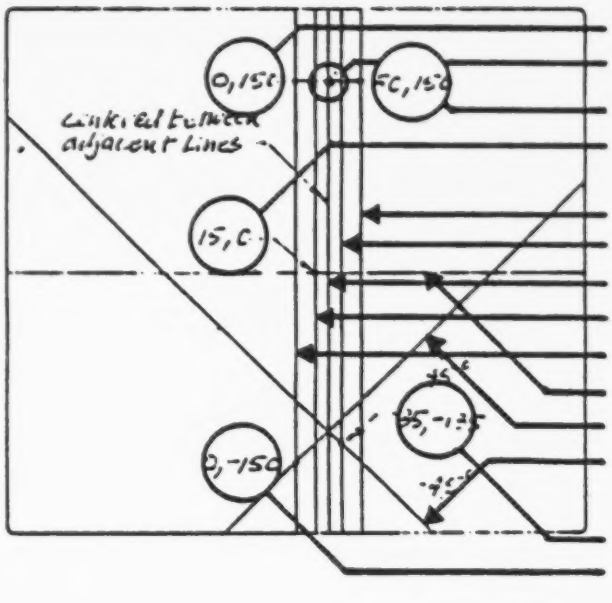
_____ 1 Logon to CADAM

_____ 2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials.

_____ 3 Go to Window Size = 15

_____ 4 Create the drawing shown to the right

_____ 5. When it looks like this, FILE the drawing



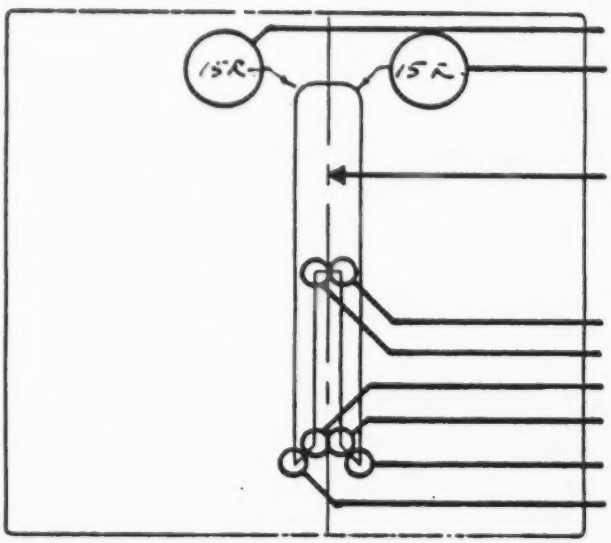
_____ 6. Call up the list of drawings.

_____ 7. Call back the drawing you just did. C.P. GRID,XXX.

_____ 8 Change the drawing as shown to the right.

_____ 9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX.

_____ 10. Logoff your User ID only.



_____ B test

_____ T test

1:21 T tutorial (C) W

Score _____/33

% _____

Subject 4 Tape 2, #1

Scoring Sheet

TEST 1

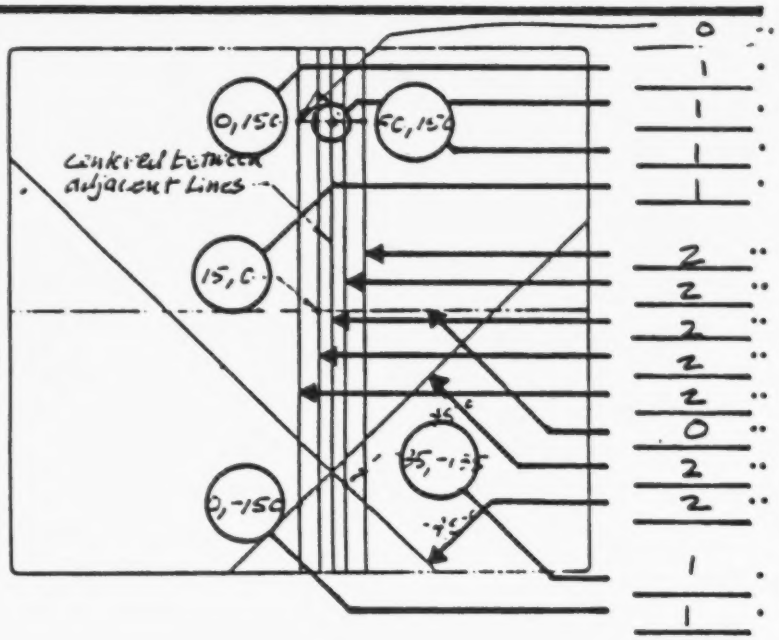
:0:42 1. Logon to CADAM

:5:08 2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials.

:8:50 3. Go to Window Size = 15 *

:16:08 4. Create the drawing shown to the right (incomplete)

:16:30 5. When it looks like this, FILE the drawing



* Default window = 15 - 1.5 would result in prints delay.

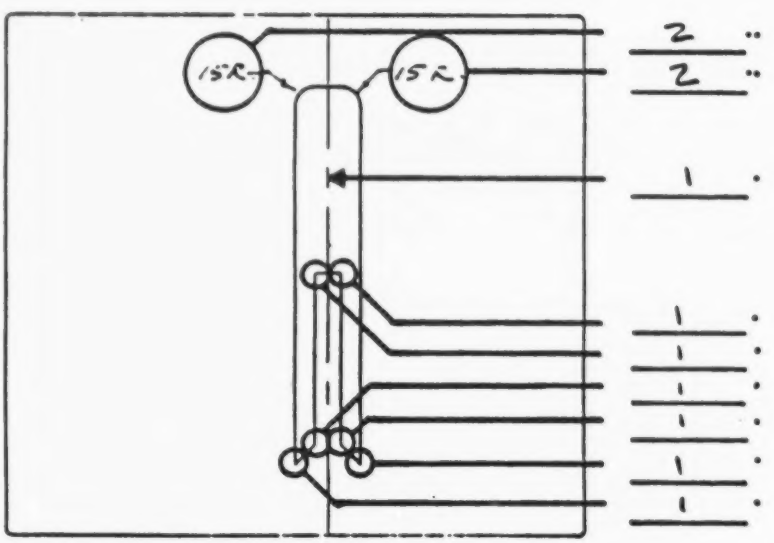
:18:23 6. Call up the list of drawings.

:18:43 7. Call back the drawing you just did. C.P. GRID,XXX.

:41:39 8. Change the drawing as shown to the right.

:42:44 9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX.

:43:08 10. Logoff your User ID only.



:0:17 B test

:42:13 T test

2:55 T tutorial C (W)

Score 31 /3535

% 89

184

Subject 5 10/8.1 #3

Scoring Sheet CADAM Hung-up @ :55:37 on test, Didn't Complete*

36 TEST 1

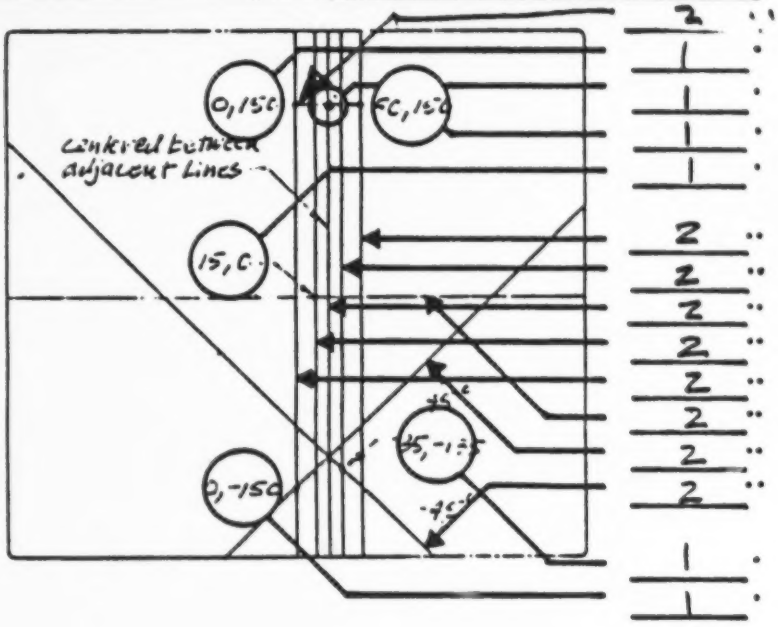
(:5:36) 1. Logon to CADAM

:0:09 2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials.

:0:44 3. Go to Window Size = 15

:25:17 4. Create the drawing shown to the right

:27:25 5. When it looks like this, FILE the drawing (Actually she logs off!)



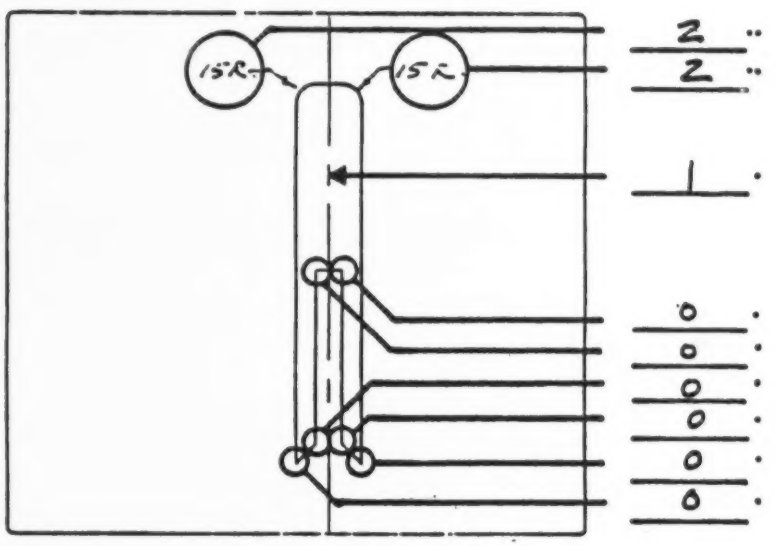
:27:53 6. Call up the list of drawings.

:28:01 7. Call back the drawing you just did. C.P. GRID,XXX.

Logs off @ :49:46 8. Change the drawing as shown to the right.

_____ 9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX.

_____ 10. Logoff your User ID only.



(:0:55) B test

:55:37 T test*

2:29: T tutorial (C) W

Clock discrepancy till 0:5:51 after she began.

Score 29 /~~33~~ 35

% 83

49 46
9 51
105
54.97

Subject 6 Tape 3, #1

Scoring Sheet Adam: Hung up in 1st try, Scores are for 2nd try (8 days later.)

TEST 2

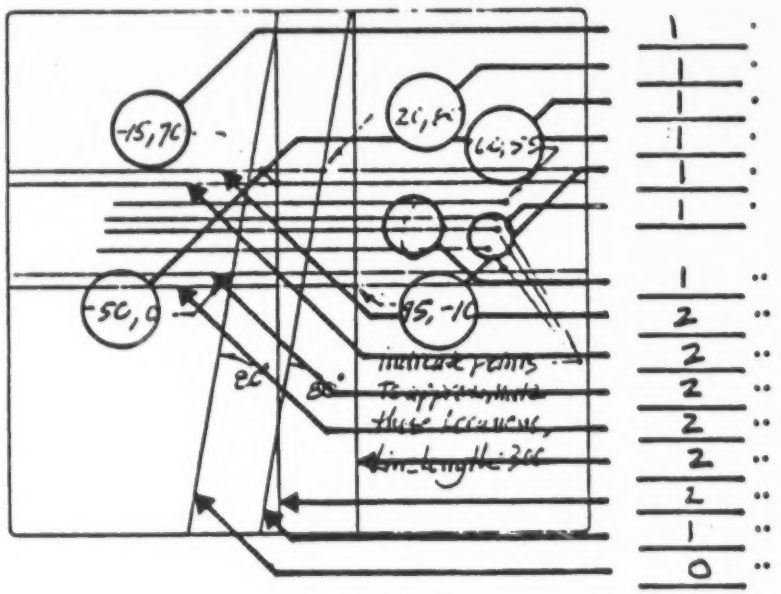
:1:54 1. Logon to CADAM.

:14:36 2 Start a drawing with the Part Number D.L. GRID and the Find Number XXX (your initials.

:15:13 3 Go to Window Size = Simple, 1.5 (4) 1:02:05

_____ 4 Create the drawing shown to the right.

_____ 5 When it looks like this, FILE the drawing.



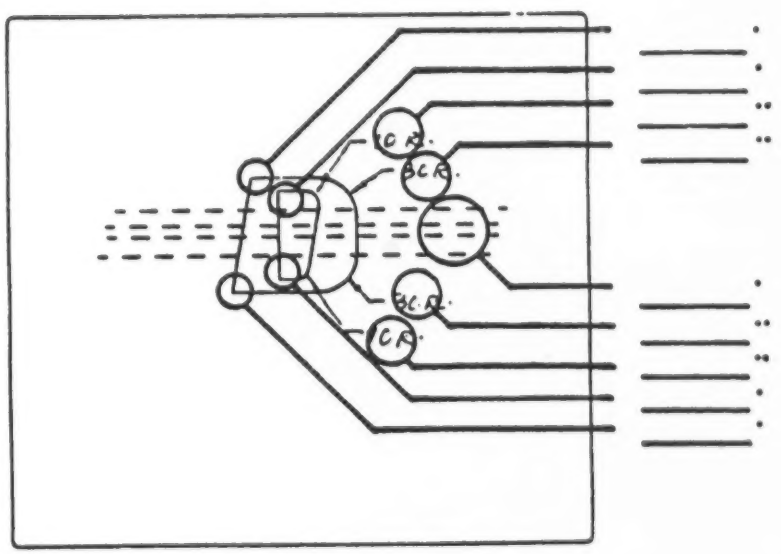
_____ 6. Call up the list of drawings.

_____ 7. Call back the drawing you just did. D.L. GRID,XXX.

_____ 8. Change the drawing as shown to the right.

_____ 9. When it looks like this, rename and file the drawing as: DASH LOGO,XXX.

_____ 10. Logoff your User ID only.



0 B test

1:00:11 T test

1:52 T tutorial (C) W

Score 20 /37

% 54

Subject 7 tq: 1 # 1

Scoring Sheet

TEST 1

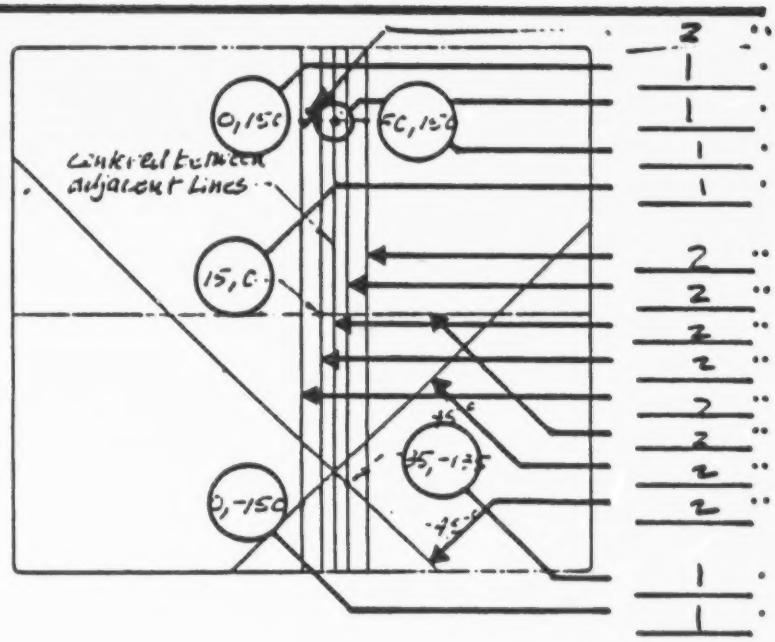
:0:20 1 Logon to CADAM

:15:57 2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials.

:18:09 3 Go to Window Size = 15

:54:15 4 Create the drawing shown to the right

55:38 5. When it looks like this, FILE the drawing



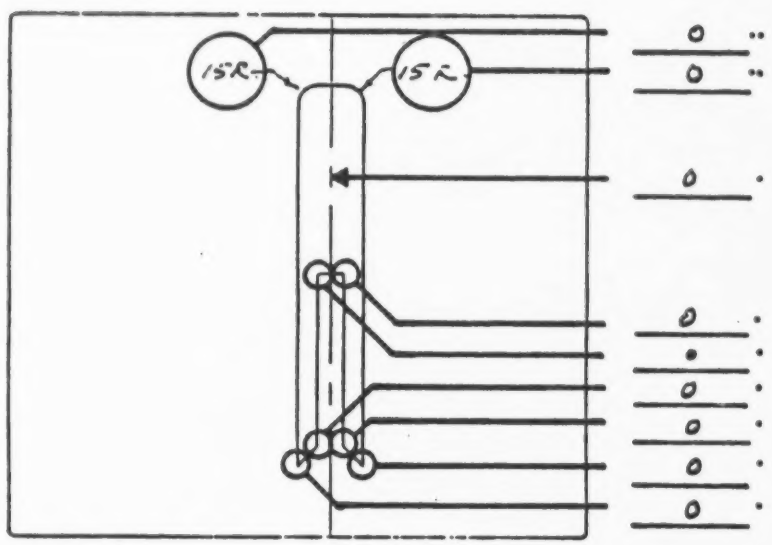
_____ 6. Call up the list of drawings.

_____ 7. Call back the drawing you just did. C.P. GRID,XXX.

_____ 8 Change the drawing as shown to the right.

_____ 9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX.

_____ 10. Logoff your User ID only.



- 0 B test

1:00:08 T test

Score 24 /35

2:30 T tutorial (C) W

% 69

Subject 8 type .1 #5

Scoring Sheet

TEST 2

:0:17 1. Logon to CADAM

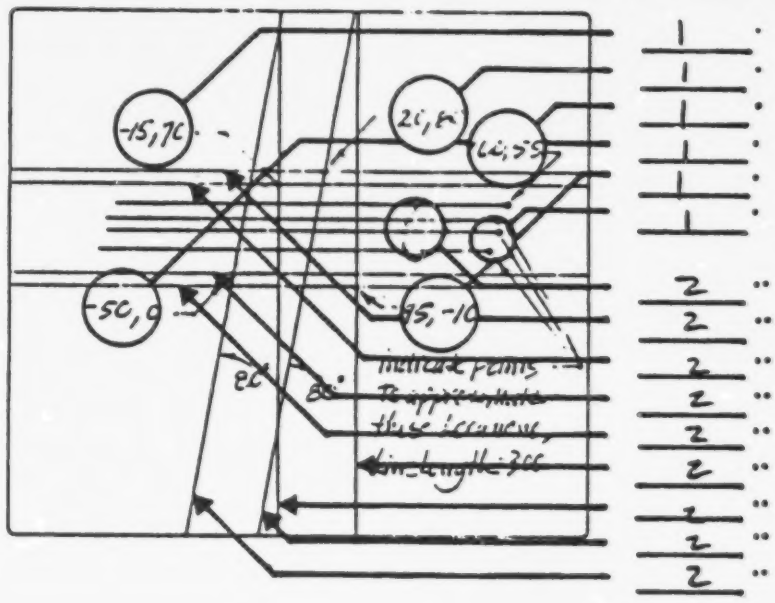
:1:15 2 Start a drawing with the Part Number D.L. GRID and the Find Number XXX (your initials.

:1:44 3 Go to Window Size = 1.5

:40:03 4 Create the drawing shown to the right.

:44:06 5 When it looks like this, FILE the drawing.

C. mrs/sks. (NO Y/N)



grmgrp @ :49:33

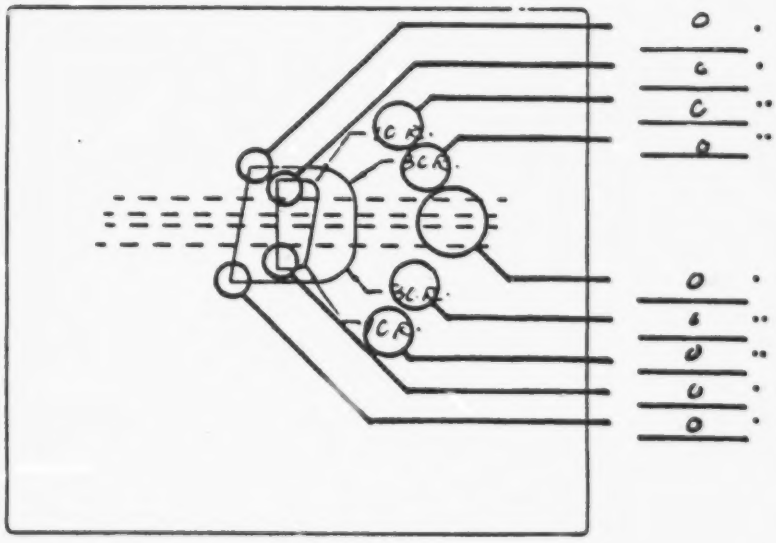
_____ 6. Call up the list of drawings.

_____ 7. Call back the drawing you just did. D.L. GRID,XXX.

_____ 8. Change the drawing as shown to the right.

_____ 9. When it looks like this, rename and file the drawing as: DASH LOGO,XXX.

_____ 10. Logoff your User ID only.



- 0 B test

0:49:33 T test

2:04 T tutorial C (W)

Score 24 /37

% 65

Subject 9 Tape 2, #2

Scoring Sheet

TEST 1

:0:32 1 Logon to CADAM

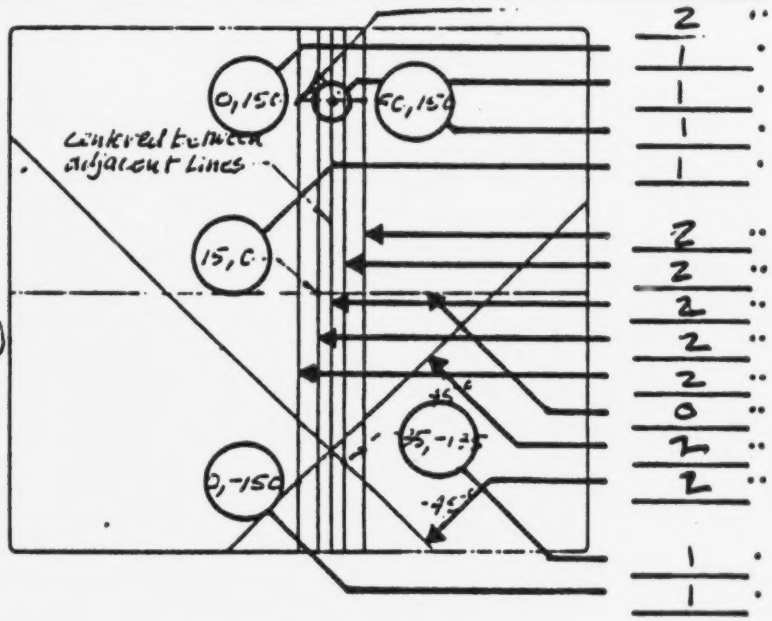
:5:20 2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials. (improper; renames old drwg.)

:6:59 3 Go to Window Size = 15

:21:37 *

158:21 4 Create the drawing shown to the right (incomplete)

:40:04 5. When it looks like this, FILE the drawing



2 1 1 1 1 2 2 2 2 2 0 2 2 1 1

* Makes drawing once, files @ :23:30
makes drawing again. files at :40:04

:44:15
Interruption Jo shift
(-0:0:11)

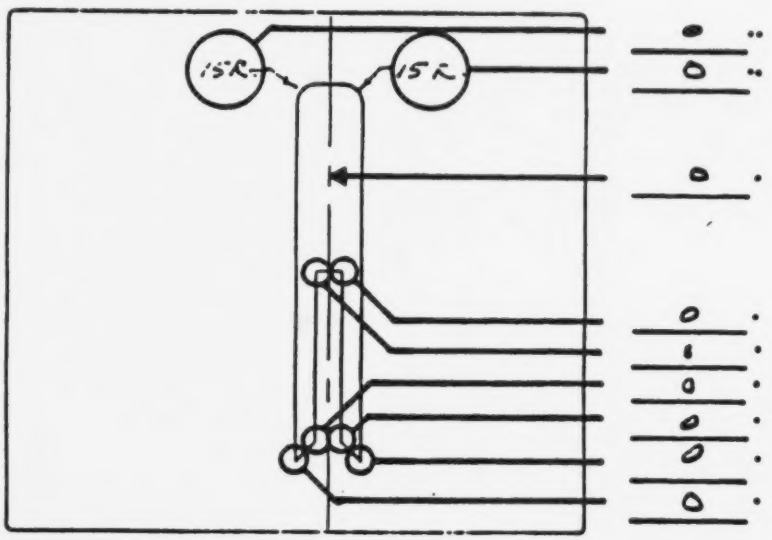
:42:30 6. Call up the list of drawings.

(-0:0:11) 7. Call back the drawing you just did. C.P. GRID,XXX.

(Skipped ← :15:40) 8 Change the drawing as shown to the right.

9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX.

10. Logoff your User ID only.



0 0 0 0 1 0 0 0 0

-:0:09 B test

1:0:07 T test

2:20 T tutorial C (W)

11
15:31
44:06
60:07

Score 22 ~~35~~

% 63

Subject 10 Tape 1 #6

Scoring Sheet

TEST 1

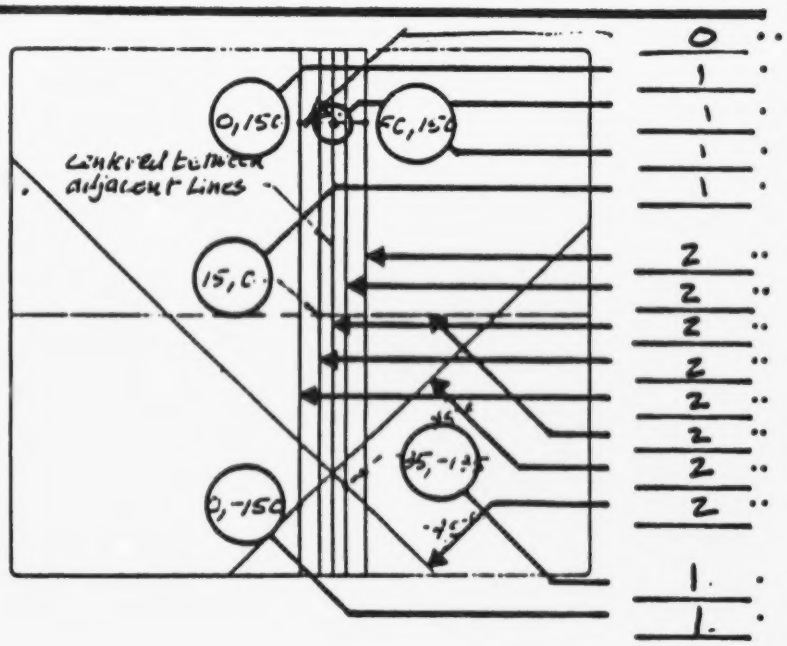
:03:50 1 Logon to CADAM

:04:09 2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials.

:05:24 3 Go to Window Size = 15

:31:15 4. Create the drawing shown to the right

:31:42 5. When it looks like this, FILE the drawing



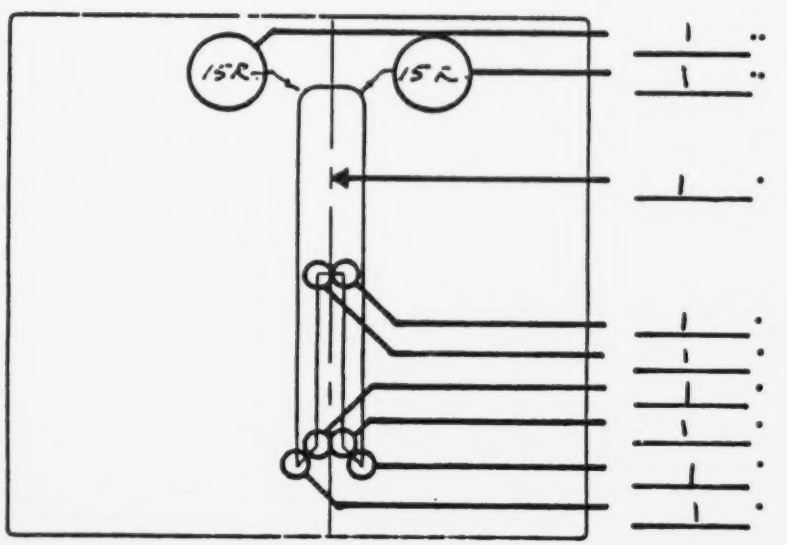
:31:53 6. Call up the list of drawings.

:32:00 7. Call back the drawing you just did. C.P. GRID,XXX.

:59:30 8. Change the drawing as shown to the right.

:59:49 9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX.

1:00:01 10. Logoff your User ID only.



:03:15 B test

:56:46 T test

1:55 T tutorial (C) W

Score 31 /35

% 89

sq t1 31: -56:46

Subject 11 Tape 3, #2

Scoring Sheet

Logs off & on again @ 8:31
Handedness again @ 9:17

TEST 1

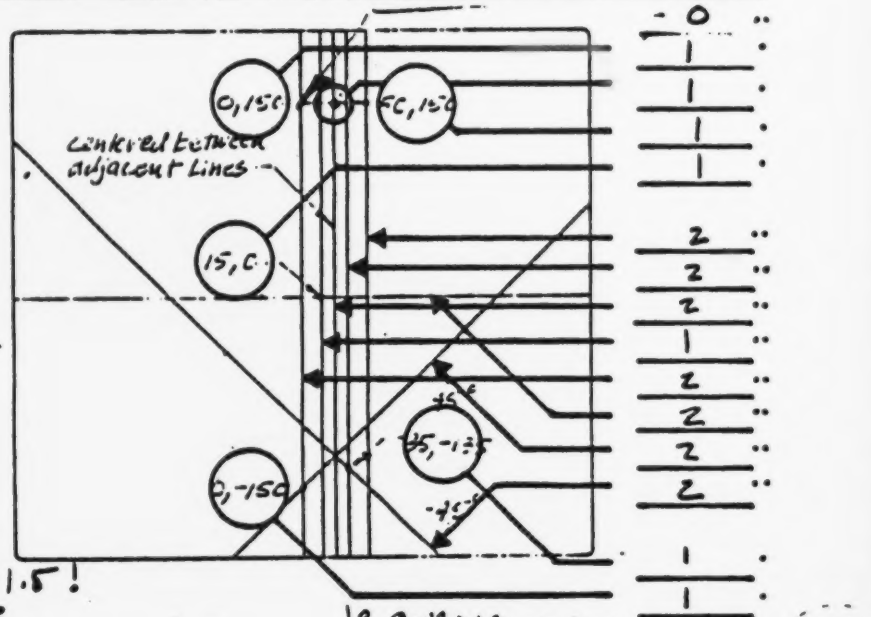
:1:59 1. Logon to CADAM

:3:01 2. Start a drawing with the Part Number C.P. GRID and the Find Number XXX (your initials.

:12:44 3. Go to Window Size = 15

:18:49 4. Create the drawing shown to the right Jacob plotter.

:19:20 5. When it looks like this, FILE the drawing

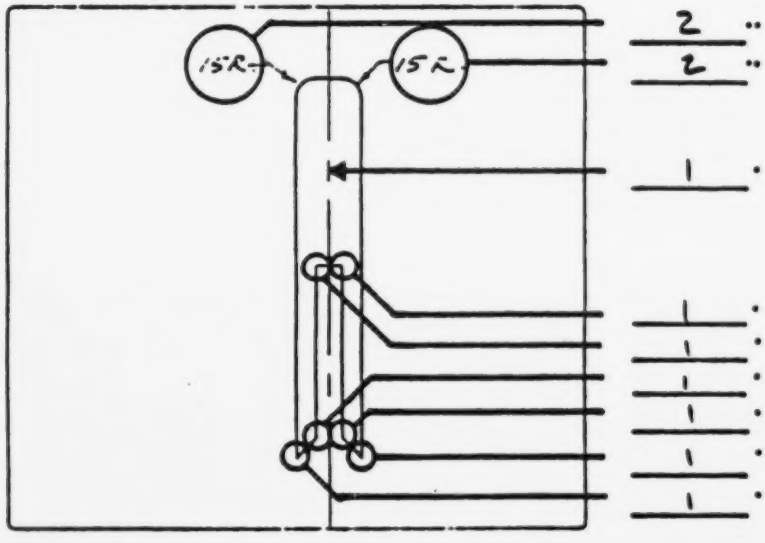


1 Can't remember is SIZE = 1.5!
2 Erases drawing by 12::38
3 Calls back again at size = 1 by :24:18
4 Logs off at :24:51 & on again at :25:03
5 Stands draw again at :25:57
6 has new drawing at size = 1.5 by :26:01
7 Files it @ 30:02
8 Calls New one Back @ size 1.0 at 30:35
9 calls back again " @ 31:48
10 grid size = 1.5 @ 32:18

:19:36 6. Call up the list of drawings

:19:42 7. Call back the drawing you just did. C.P. GRID,XXX

:38:47 8. Change the drawing as shown to the right.



:39:27 9. When it looks like this, rename and file the drawing as: CLOTHES PIN,XXX. misfiled.

:39:47 10. Logoff your User ID only.

-:1:49 B test

:37:48 T test

1:28 T tutorial C (W)

Score 32 ~~33~~ 35

% 91

37:48

Subject 12

Scoring Sheet

TEST 2

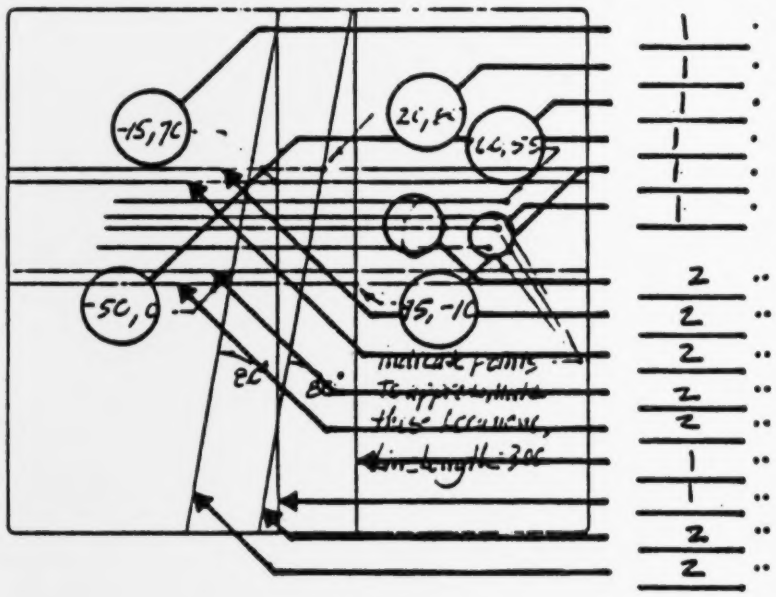
:0:08 1. Logon to CADAM.
logs off twice more.

:9:51 2. Start a drawing with the Part Number D.L. GRID and the Find Number XXX (your initials.

:10:04 3. Go to Window Size = 1.5

:33:57 4. Create the drawing shown to the right.

Did not file 5. When it looks like this, FILE the drawing.



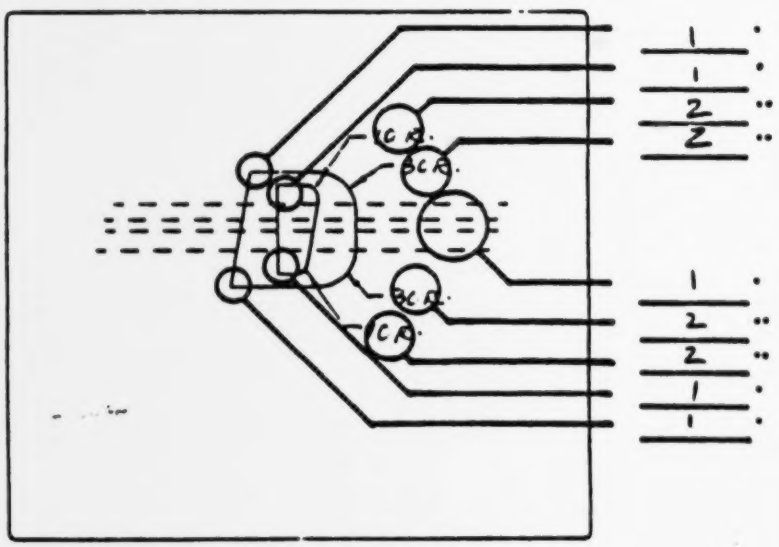
did not create 6. Call up the list of drawings.

did not recall 7. Call back the drawing you just did. D.L. GRID,XXX.

:42:28 8. Change the drawing as shown to the right.

drawing file 9. When it looks like this, rename and file the drawing as: DASH LOGO,XXX.

did not logoff 10. Logoff your User ID only.



(0:0:04) B test

:42:20 T test

1:30 T tutorial C (W)

Score 35 /37

% 95